P versus NP over Various Structures

Klaus Meer

Brandenburg University of Technology, Cottbus, Germany

26th ESSLLI Summer School 2014, Tübingen, August 2014

PART 4: Real versions of classical problems

jointly with J.A. Makowsky

Outline today

1. Introduction
2. Transfer results
3. Inside $\mathrm{NP}_{\mathbb{R}}$
4. Recursion theory over $\mathbb{R}$

1. Introduction: Recall basics in BSS complexity theory

BSS model of computability and complexity over $\mathbb{R}$ and $\mathbb{C}$:

Algorithms allow as basic steps arithmetic operations $+, -, \bullet$ as well as test-operation: $\geq$ over $\mathbb{R}$ and $=$ over $\mathbb{C}$

1. Introduction: Recall basics in BSS complexity theory

BSS model of computability and complexity over $\mathbb{R}$ and $\mathbb{C}$:

Algorithms allow as basic steps arithmetic operations $+, -, \bullet$ as well as test-operation: $\geq$ over $\mathbb{R}$ and $=$ over $\mathbb{C}$

Decision problem:	$L \subseteq \mathbb{R}^* := \bigsqcup_{n \geq 1} \mathbb{R}^n$
Size of problem instance:	number of reals specifying input
Cost of an algorithm:	number of operations

Important: Algorithms are allowed to introduce finite set of parameters into its calculations: Machine constants

Definition (Complexity class $P_{\mathbb{R}}$)

$L \in P_{\mathbb{R}}$ if efficiently decidable, i.e., number of steps in an algorithm deciding whether input $x \in \mathbb{R}^*$ belongs to L polynomially bounded in (algebraic) size of input x

Example

Solvability of linear system $A \cdot x = b$ by Gaussian elimination;

Existence of real solution of univariate polynomial $f \in \mathbb{R}[x]$

Definition (Complexity class $\mathrm{P}_{\mathbb{R}}$)

$L \in \mathrm{P}_{\mathbb{R}}$ if efficiently decidable, i.e., number of steps in an algorithm deciding whether input $x \in \mathbb{R}^*$ belongs to L polynomially bounded in (algebraic) size of input x

Example

Solvability of linear system $A \cdot x = b$ by Gaussian elimination;

Existence of real solution of univariate polynomial $f \in \mathbb{R}[x]$

(Sturm)

Definition (Complexity class $\mathrm{NP}_{\mathbb{R}}$)

$L \in \mathrm{NP}_{\mathbb{R}}$ if efficiently verifiable, i.e., given $x \in \mathbb{R}^*$ and potential membership proof $y \in \mathbb{R}^*$, there is an algorithm verifying whether y proves $x \in L$.

Definition (Complexity class $\mathrm{NP}_{\mathbb{R}}$)

$L \in \mathrm{NP}_{\mathbb{R}}$ if efficiently verifiable, i.e., given $x \in \mathbb{R}^*$ and potential membership proof $y \in \mathbb{R}^*$, there is an algorithm verifying whether y proves $x \in L$.

If $x \in L$ there must exist such a proof; if $x \notin L$ no proof y is accepted.

Definition (Complexity class $\mathrm{NP}_{\mathbb{R}}$)

$L \in \mathrm{NP}_{\mathbb{R}}$ if efficiently verifiable, i.e., given $x \in \mathbb{R}^*$ and potential membership proof $y \in \mathbb{R}^*$, there is an algorithm verifying whether y proves $x \in L$.

If $x \in L$ there must exist such a proof; if $x \notin L$ no proof y is accepted.

The running time is polynomially bounded in (algebraic) size of input x (and thus, only polynomially bounded y's are relevant)

Example

1.) Quadratic Polynomial Systems QPS (Hilbert Nullstellensatz):

Input: $n, m \in \mathbb{N}$, real polynomials in n variables

$p_1, \ldots, p_m \in \mathbb{R}[x_1, \ldots, x_n]$ of degree at most 2; each p_i depending on at most 3 variables;

Do the p_i's have a common real root?

Example

1.) Quadratic Polynomial Systems QPS (Hilbert Nullstellensatz):

Input: $n, m \in \mathbb{N}$, real polynomials in n variables

$p_1, \ldots, p_m \in \mathbb{R}[x_1, \ldots, x_n]$ of degree at most 2; each p_i depending on at most 3 variables;

Do the p_i's have a common real root?

$\mathrm{NP}_{\mathbb{R}}$-verification for solvability of system

$$p_1(x) = 0 \ , \ \ldots, p_m(x) = 0$$

guesses solution $y^* \in \mathbb{R}^n$ and plugs it into all p_i's ; obviously all components of y^* have to be seen

Definition ($\mathrm{NP}_{\mathbb{R}}$-completeness)

L is $\mathrm{NP}_{\mathbb{R}}$-complete if each problem A in $\mathrm{NP}_{\mathbb{R}}$ can be reduced in polynomial time to L, i.e., instead of deciding whether $x \in A$ one can decide whether $f(x) \in L$, where f can be computed in polynomial time in $size_{\mathbb{R}}(x)$.

Definition ($\mathrm{NP}_{\mathbb{R}}$-completeness)

L is $\mathrm{NP}_{\mathbb{R}}$-complete if each problem A in $\mathrm{NP}_{\mathbb{R}}$ can be reduced in polynomial time to L, i.e., instead of deciding whether $x \in A$ one can decide whether $f(x) \in L$, where f can be computed in polynomial time in $size_{\mathbb{R}}(x)$.

Complete problems have universal complexity within $\mathrm{NP}_{\mathbb{R}}$

Main open problem: Is $\mathrm{P}_{\mathbb{R}} = \mathrm{NP}_{\mathbb{R}}$?

Equivalent: Are there $\mathrm{NP}_{\mathbb{R}}$-complete problems in $\mathrm{P}_{\mathbb{R}}$?

Definition ($\mathrm{NP}_{\mathbb{R}}$-completeness)

L is $\mathrm{NP}_{\mathbb{R}}$-complete if each problem A in $\mathrm{NP}_{\mathbb{R}}$ can be reduced in polynomial time to L, i.e., instead of deciding whether $x \in A$ one can decide whether $f(x) \in L$, where f can be computed in polynomial time in $size_{\mathbb{R}}(x)$.

Complete problems have universal complexity within $\mathrm{NP}_{\mathbb{R}}$

Main open problem: Is $\mathrm{P}_{\mathbb{R}} = \mathrm{NP}_{\mathbb{R}}$?

Equivalent: Are there $\mathrm{NP}_{\mathbb{R}}$-complete problems in $\mathrm{P}_{\mathbb{R}}$?

Remark.

Similar definitions for structures like $\mathbb{C}$ (with $=?$ test), groups, vector spaces, ...

Theorem (Blum-Shub-Smale '89)

a) *The Hilbert-Nullstellensatz problem* $QPS_{\mathbb{R}}$ *is* $NP_{\mathbb{R}}$*-complete. Considered as problem* $QPS_{\mathbb{C}}$ *over* $\mathbb{C}$ *it is* $NP_{\mathbb{C}}$*-complete.*

b) *The real Halting problem* $\mathbb{H}_{\mathbb{R}}$ *is undecidable in the BSS model: Given a machine* M *(as codeword in* $\mathbb{R}^*$*) together with input* $x \in \mathbb{R}^*$*, does* M *halt on* x*?*

c) *Other undecidable problems:* $\mathbb{Q}$ *inside* $\mathbb{R}$*, the Mandelbrot set as subset of* $\mathbb{R}^2$

Theorem (Blum-Shub-Smale '89)

a) *The Hilbert-Nullstellensatz problem* $QPS_{\mathbb{R}}$ *is* $NP_{\mathbb{R}}$*-complete. Considered as problem* $QPS_{\mathbb{C}}$ *over* $\mathbb{C}$ *it is* $NP_{\mathbb{C}}$*-complete.*

b) *The real Halting problem* $\mathbb{H}_{\mathbb{R}}$ *is undecidable in the BSS model: Given a machine* M *(as codeword in* $\mathbb{R}^*$*) together with input* $x \in \mathbb{R}^*$*, does* M *halt on* x*?*

c) *Other undecidable problems:* $\mathbb{Q}$ *inside* $\mathbb{R}$*, the Mandelbrot set as subset of* $\mathbb{R}^2$

Both $\mathbb{H}_{\mathbb{R}}$ and $\mathbb{Q}$ are semi-decidable, i.e., there is a BSS algorithm that halts precisely on inputs from these sets.

Theorem

All problems in $\mathrm{NP}_{\mathbb{R}}$ are decidable in simple exponential time; similarly for $NP_{\mathbb{C}}$.

Theorem

All problems in $\mathrm{NP}_{\mathbb{R}}$ are decidable in simple exponential time; similarly for $NP_{\mathbb{C}}$.

Proof.

Difficulty: uncountable search space; requires quantifier elimination algorithms for real/algebraically closed fields

Long history starting with Tarski; fundamental contributions by Collins, Heintz et al., Grigoriev & Vorobjov, Renegar, Basu & Pollack & Roy, ...

Effective Hilbert Nullstellensatz: Giusti & Heintz, Pardo, ...

General guideline for topics treated below:

What about important questions and results in the Turing model when studied in new computational frameworks?

- P versus NP question?
- impact of results in new models for Turing model?
- and vice versa?
- benefit of different (mathematical) methods available for studying computability on a structure, for example, separation of complexity classes?
- impact of using real or complex constants in an algorithm?
- ...

We shall see that all kind of answers occur such as

- (almost) trivial transfer of similar statements:

  $\mathrm{NP}_{\mathbb{R}}$-completeness of $\mathrm{HNS}_{\mathbb{R}}$ problem; undecidability of $\mathbb{H}_{\mathbb{R}}$
- deep results concerning transfer of P versus NP results
- new framework sheds light as well on Turing results; new interesting questions arise
- difficult problems in Turing setting have easier real answers and vice versa
- etc.

Problems treated in this and the next talk:

1. Transfer results

Problems treated in this and the next talk:

1. Transfer results
2. Ladner's theorem concerning structure inside NP: Are there non-complete problems between P and NP?

   depending on structure more difficult, new open problems

Problems treated in this and the next talk:

1. Transfer results
2. Ladner's theorem concerning structure inside NP: Are there non-complete problems between P and NP?

   depending on structure more difficult, new open problems
3. Recursion theory: Undecidable problems, degrees of undecidability

   some results easier: Post's problem

Problems treated in this and the next talk:

1. Transfer results
2. Ladner's theorem concerning structure inside NP: Are there non-complete problems between P and NP?

   depending on structure more difficult, new open problems
3. Recursion theory: Undecidable problems, degrees of undecidability

   some results easier: Post's problem
4. Characterization of $\mathrm{NP}_{\mathbb{R}}$ through PCPs

   new problems for algebraic computations

4. Transfer results

Since P versus NP is major question in above (and further) models as well it is natural to ask, how these (and further) questions relate in different models, in particular:

how is classical Turing complexity theory related to results over $\mathbb{R}, \mathbb{C}, \ldots$?

Transfer Results

Theorem (Blum & Cucker & Shub & Smale 1996)

For all algebraically closed fields of characteristic 0 *the P versus NP question has the same answer.*

Theorem (Blum & Cucker & Shub & Smale 1996)

For all algebraically closed fields of characteristic 0 the P versus NP question has the same answer.

Proof.

Main idea is to eliminate complex machine constants in algorithms for problems that can be defined without such constants;

the $\mathrm{NP}_{\mathbb{C}}$-complete problem QPS has this property;

price to pay for elimination only polynomial slowdown

Technique: Some algebraic number theory

Elimination of machine constants important technique for several transfer results; alternative proof by Koiran does it applying again Quantifier Elimination:

- algebraic constants are coded via minimal polynomials
- transcendental constants satisfy no algebraic equality test in algorithm, so each test is answered the same in a neighborhood of such a constant; using deep results from complex QE shows that there is a small rational point in such a neighborhood which can replace the transcendental constant. It can be computed without performing the QE.

Relation between complex BSS model and randomized Turing algorithms through class BPP of discrete problems that can be decided with small two-sided error in polynomial time

Relation between complex BSS model and randomized Turing algorithms through class BPP of discrete problems that can be decided with small two-sided error in polynomial time

Theorem (Smale, Koiran)

Suppose $P_{\mathbb{C}} = NP_{\mathbb{C}}$*, then* $\mathrm{NP} \subseteq \mathrm{BPP}$.

Relation between complex BSS model and randomized Turing algorithms through class BPP of discrete problems that can be decided with small two-sided error in polynomial time

Theorem (Smale, Koiran)

Suppose $P_{\mathbb{C}} = NP_{\mathbb{C}}$*, then* $\mathrm{NP} \subseteq \mathrm{BPP}$.

Proof.

Extract from $\mathrm{P}_{\mathbb{C}}$ algorithm for $\mathrm{QPS}_{\mathbb{C}}$ a randomized algorithm for NP-complete variant of QPS;

Relation between complex BSS model and randomized Turing algorithms through class BPP of discrete problems that can be decided with small two-sided error in polynomial time

Theorem (Smale, Koiran)

Suppose $P_{\mathbb{C}} = NP_{\mathbb{C}}$, then $\mathrm{NP} \subseteq \mathrm{BPP}$*.*

Proof.

Extract from $\mathrm{P}_{\mathbb{C}}$ algorithm for $\mathrm{QPS}_{\mathbb{C}}$ a randomized algorithm for NP-complete variant of QPS; replacement of complex constants by randomly choosing small rational constants from a suitable set which with high probability contains rationals that behave the same as original constants.

Both above results not known for real algorithms; first deeper relation between real and Turing algorithms via additive real BSS machines, i.e., algorithms that only perform $+, -$ and tests $x \geq 0$;

Both above results not known for real algorithms; first deeper relation between real and Turing algorithms via additive real BSS machines, i.e., algorithms that only perform $+, -$ and tests $x \geq 0$; here: no non-rational machine constants

Both above results not known for real algorithms; first deeper relation between real and Turing algorithms via additive real BSS machines, i.e., algorithms that only perform $+, -$ and tests $x \geq 0$; here: no non-rational machine constants

Theorem (Fournier & Koiran 1998)

$$\mathrm{P} = \mathrm{NP} \textit{ (Turing)} \Leftrightarrow P_{\mathbb{R}}^{add} = NP_{\mathbb{R}}^{add} \textit{ (additive model)}$$

Both above results not known for real algorithms; first deeper relation between real and Turing algorithms via additive real BSS machines, i.e., algorithms that only perform $+, -$ and tests $x \geq 0$;

here: no non-rational machine constants

Theorem (Fournier & Koiran 1998)

$$\mathrm{P} = \mathrm{NP} \textit{ (Turing)} \Leftrightarrow P_{\mathbb{R}}^{add} = NP_{\mathbb{R}}^{add} \textit{ (additive model)}$$

Proof.

Replacement of machine constants using deep result on point location in hyperplane arrangements by Meyer auf der Heide

Remark.

1. Similar results when allowing real machine constants, but introduces non-uniformity into Turing results.
2. In additive model with equality tests only, P and NP are provably different; however, questions about the polynomial hierarchy in this setting are as difficult as major open questions in classical complexity (Koiran).

3. Inside $\mathrm{NP}_{\mathbb{R}}$: Ladner's theorem in different contexts

Classical result in Turing complexity/recursion theory:

3. Inside $\mathrm{NP}_{\mathbb{R}}$: Ladner's theorem in different contexts

Classical result in Turing complexity/recursion theory:

Theorem (Ladner 1975)

If $\mathrm{P} \neq \mathrm{NP}$ there are non-complete problems in $\mathrm{NP} \setminus \mathrm{P}$.

3. Inside $\mathrm{NP}_{\mathbb{R}}$: Ladner's theorem in different contexts

Classical result in Turing complexity/recursion theory:

Theorem (Ladner 1975)

If $\mathrm{P} \neq \mathrm{NP}$ *there are non-complete problems in* $\mathrm{NP} \setminus \mathrm{P}$.

Proof.

Key point is diagonalization against family

$\{P_1, P_2, \ldots\}, P_i = (M_i, p_i)$ of decision machines M_i with

polynomial time bound p_i and family $\{R_1, R_2, \ldots\}, R_i = (N_i, q_i)$ of

reduction machines N_i with polynomial time bound q_i;

both families are countable and effectively enumerable in Turing

model;

Proof (cntd.)

given NP-complete $3SAT$ construct $L \in \mathrm{NP}$ s.t. one after the other for increasing i M_i fails to decide L within time bound p_i and N_i fails to reduce $3SAT$ to L within time bound q_i;

Proof (cntd.)

given NP-complete $3SAT$ construct $L \in \mathrm{NP}$ s.t. one after the other for increasing i M_i fails to decide L within time bound p_i and N_i fails to reduce $3SAT$ to L within time bound q_i;

L constructed as dimensionwise variation of 3SAT: decompose $\mathbb{N} = S \cup \bar{S}$ such that

- for inputs with length/dimension $n \in S$ L is defined as empty set and thus an easy problem;
- for inputs with length $n \in \bar{S}$ L is defined as 3SAT and thus difficult

Proof (cntd.)

Clear: if L looks like $\emptyset$ a machine from $\{R_i\}$ finally errs, if L looks like 3SAT a machine from $\{P_i\}$ finally errs

Proof (cntd.)

Clear: if L looks like $\emptyset$ a machine from $\{R_i\}$ finally errs, if L looks like 3SAT a machine from $\{P_i\}$ finally errs

Heart of the proof: define set S appropriately by finding stepwise and effectively error dimensions for $P_i, R_i, i = 1, 2, \ldots$

Proof (cntd.)

Clear: if L looks like $\emptyset$ a machine from $\{R_i\}$ finally errs, if L looks like 3SAT a machine from $\{P_i\}$ finally errs

Heart of the proof: define set S appropriately by finding stepwise and effectively error dimensions for $P_i, R_i, i = 1, 2, \ldots$

Idea: start to fool P_1 by defining $L = 3SAT$ on dimensions $1, 2, \ldots, n_1$; here n_1 should be large enough such that there is an input formula of length at most n_1 which is decided falsely by P_1 within at most $p_1(n_1)$ steps;

Proof (cntd.)

Clear: if L looks like $\emptyset$ a machine from $\{R_i\}$ finally errs, if L looks like 3SAT a machine from $\{P_i\}$ finally errs

Heart of the proof: define set S appropriately by finding stepwise and effectively error dimensions for $P_i, R_i, i = 1, 2, \ldots$

Idea: start to fool P_1 by defining $L = 3SAT$ on dimensions $1, 2, \ldots, n_1$; here n_1 should be large enough such that there is an input formula of length at most n_1 which is decided falsely by P_1 within at most $p_1(n_1)$ steps; do the same with R_1 finding an $n_2 > n_1$ and defining $L = \emptyset$ on dimensions $\in (n_1, \ldots, n_2]$; etc.

rest a folklore padding argument to force L into NP

Blum-Shub-Smale model over $\mathbb{R}, \mathbb{C}$: set of algorithms uncountable

thus, direct transformation of above construction fails

Blum-Shub-Smale model over $\mathbb{R}, \mathbb{C}$: set of algorithms uncountable

thus, direct transformation of above construction fails

Turns out to be surprisingly interesting question:

- over $\mathbb{C}$ positive answer because of transfer theorem or, alternatively, model theoretic considerations; sheds also more light on why classical proof works
- over $\mathbb{R}$ surprisingly difficult and so far unsolved; partial results for restricted real models known
- leads to new research questions

Theorem (Malajovich & M. 1995)

If $P_{\mathbb{C}} \neq NP_{\mathbb{C}}$ there are non-complete problems in $NP_{\mathbb{C}} \setminus P_{\mathbb{C}}$.

Theorem (Malajovich & M. 1995)

If $P_{\mathbb{C}} \neq NP_{\mathbb{C}}$ there are non-complete problems in $NP_{\mathbb{C}} \setminus P_{\mathbb{C}}$.

Proof.

Transfer result by Shub & Smale: 'P = NP ?' has the same answer for all algebraically closed fields of characteristic zero; efficient elimination of complex machine constants from algorithms that deal with QPS problem allows to reduce problem to the algebraic closure $\bar{\mathbb{Q}}$ of $\mathbb{Q}$ in $\mathbb{C}$, i.e., to a countable setting; then adapt Ladner's proof

'Elimination-of-constants' result is not known for $\mathbb{R}$, so what can be done?

'Elimination-of-constants' result is not known for $\mathbb{R}$, so what can be done?

Central complexity class for investigations: $\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ (Michaux)

$\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ allows diagonalization technique in uncountable settings

idea: consider discrete skeleton of real/complex algorithms,

split real/complex constants from skeleton

'Elimination-of-constants' result is not known for $\mathbb{R}$, so what can be done?

Central complexity class for investigations: $\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ (Michaux)

$\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ allows diagonalization technique in uncountable settings

idea: consider discrete skeleton of real/complex algorithms, split real/complex constants from skeleton

$\rightsquigarrow$ basic machine:

'Elimination-of-constants' result is not known for $\mathbb{R}$, so what can be done?

Central complexity class for investigations: $\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ (Michaux)

$\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ allows diagonalization technique in uncountable settings

idea: consider discrete skeleton of real/complex algorithms,

split real/complex constants from skeleton

$\rightsquigarrow$ basic machine: $\underbrace{M}_{\text{skeleton}}$

'Elimination-of-constants' result is not known for $\mathbb{R}$, so what can be done?

Central complexity class for investigations: $\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ (Michaux)

$\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ allows diagonalization technique in uncountable settings

idea: consider discrete skeleton of real/complex algorithms, split real/complex constants from skeleton

$$\rightsquigarrow \text{basic machine: } \underbrace{M}_{\text{skeleton}} (\underbrace{x}_{input},$$

'Elimination-of-constants' result is not known for $\mathbb{R}$, so what can be done?

Central complexity class for investigations: $\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ (Michaux)

$\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ allows diagonalization technique in uncountable settings

idea: consider discrete skeleton of real/complex algorithms, split real/complex constants from skeleton

$$\rightsquigarrow \text{basic machine: } \underbrace{M}_{\text{skeleton}} \; (\underbrace{x}_{\textit{input}}, \underbrace{c}_{\substack{\textit{machine}\\ \textit{constants}}})$$

$L \in \mathrm{P}_{\mathbb{R}}/\mathrm{const} \;\Leftrightarrow$ there is a skeleton M using k constants such that

$L \in \mathrm{P}_{\mathbb{R}}/\mathrm{const} \;\Leftrightarrow$ there is a skeleton M using k constants such that for each input dimension n there is a choice $c^{(n)} \in \mathbb{R}^k$ such that $M(\bullet, c^{(n)})$ decides L upto dimension n in polynomial time.

$L \in \mathrm{P}_{\mathbb{R}}/\mathrm{const} \ \Leftrightarrow$ there is a skeleton M using k constants such that for each input dimension n there is a choice $c^{(n)} \in \mathbb{R}^k$ such that $M(\bullet, c^{(n)})$ decides L upto dimension n in polynomial time.

Important:

skeleton is used uniformly, machine constants non-uniformly,

$L \in \mathrm{P}_{\mathbb{R}}/\mathrm{const} \Leftrightarrow$ there is a skeleton M using k constants such that for each input dimension n there is a choice $c^{(n)} \in \mathbb{R}^k$ such that $M(\bullet, c^{(n)})$ decides L upto dimension n in polynomial time.

Important:

skeleton is used uniformly, machine constants non-uniformly,

$\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ is a restricted version of non-uniform class $\mathrm{P}_{\mathbb{R}}/poly$;

set of basic machines countable!

Similarly for other models: $\mathrm{P}_{\mathbb{C}}/\mathrm{const}, \mathrm{P}_{\mathbb{R}}^{\mathrm{add}}/\mathrm{const}, \mathrm{P}_{\mathbb{R}}^{\mathrm{rc}}/\mathrm{const}$

Theorem (Ben-David & M. & Michaux 2000)

If $\mathrm{NP}_{\mathbb{R}} \not\subseteq \mathrm{P}_{\mathbb{R}}/\mathrm{const}$ *there exist problems in* $\mathrm{NP}_{\mathbb{R}} \setminus \mathrm{P}_{\mathbb{R}}/\mathrm{const}$ *which are not* $\mathrm{NP}_{\mathbb{R}}$*-complete under* $\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ *reductions.*

Theorem (Ben-David & M. & Michaux 2000)

If $\mathrm{NP}_{\mathbb{R}} \not\subseteq \mathrm{P}_{\mathbb{R}}/\mathrm{const}$ *there exist problems in* $\mathrm{NP}_{\mathbb{R}} \setminus \mathrm{P}_{\mathbb{R}}/\mathrm{const}$ *which are not* $\mathrm{NP}_{\mathbb{R}}$*-complete under* $\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ *reductions.*

Proof.

Construct again diagonal problem L along Ladner's line;

fool step by step all basic decision / reduction machines;

fooling dimensions computed via quantifier elimination: for each n and basic machine M running in polynomial time it is first order expressible whether M with some choice of constants decides problem upto dimension n.

Thus central: analysis of $\mathrm{P}/\mathrm{const}$ in different models;

here notions from model theory enter

Theorem (Michaux; Ben-David & Michaux & M.)

For every ω-saturated structure it is $\mathrm{P} = \mathrm{P}/\mathrm{const}$.

Thus central: analysis of $\mathrm{P}/\mathrm{const}$ in different models;

here notions from model theory enter

Theorem (Michaux; Ben-David & Michaux & M.)

For every ω-saturated structure it is $\mathrm{P} = \mathrm{P}/\mathrm{const}$.

ω-saturation roughly means: given countable family $\phi_n(c)$ of first-order formulas such that each finite subset is commonly satisfiable, then the entire family is satisfiable.

$\mathbb{R}$ is not ω-saturated: $\phi_n(c) \equiv c \geq n$

What is known about $\mathrm{P}/\mathrm{const}$ in different models:

What is known about $\mathrm{P}/\mathrm{const}$ in different models:

Turing: $\mathrm{P} = \mathrm{P}/\mathrm{const}$ (trivial!) thus Ladner holds

What is known about $\mathrm{P}/\mathrm{const}$ in different models:

Turing: $\mathrm{P} = \mathrm{P}/\mathrm{const}$ (trivial!) thus Ladner holds

BSS over $\mathbb{C}$: $\mathrm{P}_{\mathbb{C}} = \mathrm{P}_{\mathbb{C}}/\mathrm{const}$ thus Ladner holds

What is known about $\mathrm{P}/\mathrm{const}$ in different models:

Turing: $\mathrm{P} = \mathrm{P}/\mathrm{const}$ (trivial!) thus Ladner holds

BSS over $\mathbb{C}$: $\mathrm{P}_{\mathbb{C}} = \mathrm{P}_{\mathbb{C}}/\mathrm{const}$ thus Ladner holds

BSS over $\mathbb{R}$: highly unlikely that $\mathrm{P}_{\mathbb{R}} = \mathrm{P}_{\mathbb{R}}/\mathrm{const}$

Chapuis & Koiran

What is known about $\mathrm{P}/\mathrm{const}$ in different models:

Turing: $\mathrm{P} = \mathrm{P}/\mathrm{const}$ (trivial!) — thus Ladner holds

BSS over $\mathbb{C}$: $\mathrm{P}_{\mathbb{C}} = \mathrm{P}_{\mathbb{C}}/\mathrm{const}$ — thus Ladner holds

BSS over $\mathbb{R}$: highly unlikely that $\mathrm{P}_{\mathbb{R}} = \mathrm{P}_{\mathbb{R}}/\mathrm{const}$

Chapuis & Koiran

additive BSS over $\mathbb{R}$: $\mathrm{P}_{\mathbb{R}}^{add} = \mathrm{P}_{\mathbb{R}}^{add}/\mathrm{const}$ — thus Ladner holds

Chapuis & Koiran

What is known about $\mathrm{P}/\mathrm{const}$ in different models:

Turing: $\mathrm{P} = \mathrm{P}/\mathrm{const}$ (trivial!) thus Ladner holds

BSS over $\mathbb{C}$: $\mathsf{P}_{\mathbb{C}} = \mathsf{P}_{\mathbb{C}}/\mathrm{const}$ thus Ladner holds

BSS over $\mathbb{R}$: highly unlikely that $\mathrm{P}_{\mathbb{R}} = \mathrm{P}_{\mathbb{R}}/\mathrm{const}$

Chapuis & Koiran

additive BSS over $\mathbb{R}$: $\mathsf{P}_{\mathbb{R}}^{add} = \mathsf{P}_{\mathbb{R}}^{add}/\mathrm{const}$ thus Ladner holds

Chapuis & Koiran

real BSS with restricted use of constants Ladner holds

M. 2012

Restricted BSS model:

restricted use of machine constants:

Restricted BSS model:

restricted use of machine constants:

input variables can be used arbitrarily; all intermediate results depend linearly on machine constants (thus no multiplication between machine constants)

$$\rightsquigarrow \quad \text{classes} \quad \mathrm{P}_{\mathbb{R}}^{\mathrm{rc}}, \quad \mathrm{NP}_{\mathbb{R}}^{\mathrm{rc}}, \quad \mathrm{P}_{\mathbb{R}}^{\mathrm{rc}}/\mathrm{const}$$

Restricted BSS model:

restricted use of machine constants:

input variables can be used arbitrarily; all intermediate results depend linearly on machine constants (thus no multiplication between machine constants)

$$\rightsquigarrow \quad \text{classes } \mathrm{P}_{\mathbb{R}}^{\mathrm{rc}}, \quad \mathrm{NP}_{\mathbb{R}}^{\mathrm{rc}}, \quad \mathrm{P}_{\mathbb{R}}^{\mathrm{rc}}/\mathrm{const}$$

Theorem

QPS is $NP_{\mathbb{R}}^{\mathrm{rc}}$-complete (under $P_{\mathbb{R}}^{\mathrm{rc}}$-reductions)

thus: restricted model closer to full BSS model than linear/additive models $\rightsquigarrow$ motivation for studying it!

Lemma

If QPS $\not\in P_{\mathbb{R}}^{\mathrm{rc}}/\mathrm{const}$*, then there are non-complete problems in* $NP_{\mathbb{R}}^{\mathrm{rc}} \setminus P_{\mathbb{R}}^{\mathrm{rc}}/\mathrm{const}$

(i.e. above Theorem by Ben-David & M. & Michaux holds in restricted model as well)

Lemma

If QPS $\notin P_{\mathbb{R}}^{\mathrm{rc}}/\mathrm{const}$, *then there are non-complete problems in* $NP_{\mathbb{R}}^{\mathrm{rc}} \setminus P_{\mathbb{R}}^{\mathrm{rc}}/\mathrm{const}$

(i.e. above Theorem by Ben-David & M. & Michaux holds in restricted model as well)

Main proof ingredient:

Quantifier elimination possible in restricted model, thus error dimensions for $\mathrm{P}_{\mathbb{R}}^{\mathrm{rc}}/\mathrm{const}$-computations as well as -reductions effectively computable.

Theorem (M. 2012)

Ladner's theorem holds in the real BSS model with restricted use of constants.

Theorem (M. 2012)

Ladner's theorem holds in the real BSS model with restricted use of constants.

Proof.

Main step is to prove equality $\mathrm{P}_{\mathbb{R}}^{\mathrm{rc}} = \mathrm{P}_{\mathbb{R}}^{\mathrm{rc}}/\mathrm{const}$; proof relies on a limit argument in affine geometry that allows replacement of non-uniform machine constants by uniform ones.

Let $L \in \mathrm{P}_{\mathbb{R}}^{\mathrm{rc}}/\mathrm{const}$, M corresponding basic machine using k real constants

$$E_n := \{c \in \mathbb{R}^k | M(\bullet, c) \text{ decides } \; L \cap \mathbb{R}^{\leq n}\}$$

i.e. E_n is set of suitable constants for all x upto dimension n

Proof (cntd.)

Using (among other arguments) König's lemma on infinite trees we can assume for all $n \in \mathbb{N}$:

- $E_{n+1} \subseteq E_n$, $\overline{E_{n+1}} \subseteq \overline{E_n}$
- $\bigcap_{n\in\mathbb{N}} E_n = \emptyset$
- $dim(E_n) = s$ for a fixed $s \geq 1$
- E_n is intersection of half-spaces and convex

Note: a point in $\bigcap_{n\in\mathbb{N}} E_n$ would finish the proof since it could be used as set of uniform constants for M to decide L

Proof (cntd.)

Goal: Find for each n efficiently and uniformly a choice $c^{(n)}$ of parameters such that $M(\bullet, c^{(n)})$ decides $L \cap \mathbb{R}^{\leq n}$

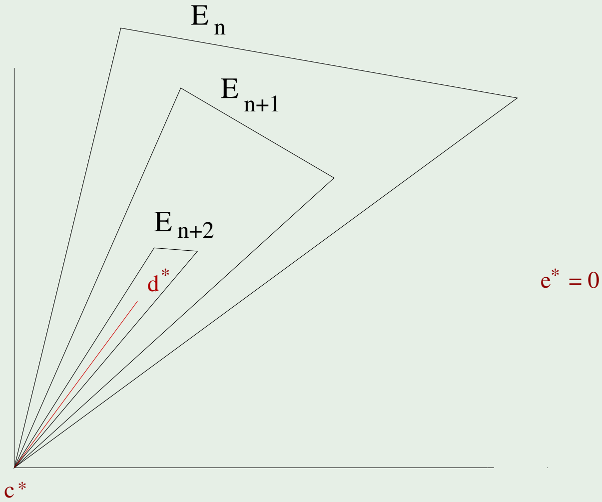
Proof (cntd.)

Goal: Find for each n efficiently and uniformly a choice $c^{(n)}$ of parameters such that $M(\bullet, c^{(n)})$ decides $L \cap \mathbb{R}^{\leq n}$

We cannot compute $c^{(n)}$, but we show existence of $c^*, d^*, e^* \in \mathbb{R}^k$ such that

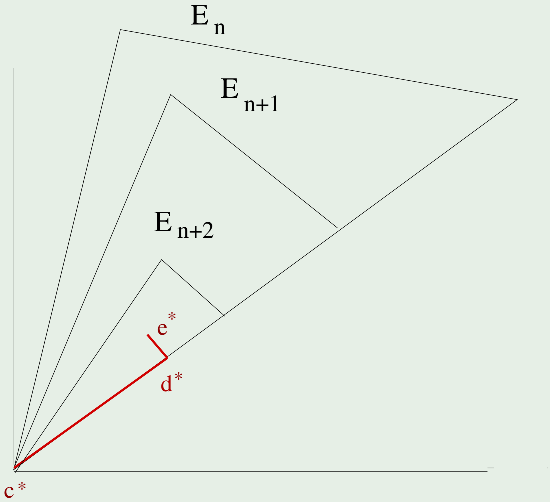
- $c^* \in \bigcap_{n \in \mathbb{N}} \overline{E_n}$
- d^* points from c^* to $\overline{E_n}$ for all n
- e^* points from $c^* + \mu_1 \cdot d^*$ to E_n for all n and $0 < \mu_1$ small enough (depending on n)

Proof (cntd.)



Limit argument: Easy case

Proof (cntd.)



Limit argument: Difficult case

Proof (cntd.)

We obtain following formula: there exist $c^*, d^*, e^* \in \mathbb{R}^k$ such that

$$\forall n \in \mathbb{N}\ \exists \epsilon_1 > 0\ \forall \mu_1 \in (0, \epsilon_1)\ \exists \epsilon_2 > 0\ \forall \mu_2 \in (0, \epsilon_2) :$$

$$c^* + \mu_1 \cdot d^* + \mu_2 \cdot e^* \in E_n\ .$$

Proof (cntd.)

We obtain following formula: there exist $c^*, d^*, e^* \in \mathbb{R}^k$ such that

$$\forall n \in \mathbb{N}\ \exists \epsilon_1 > 0\ \forall \mu_1 \in (0, \epsilon_1)\ \exists \epsilon_2 > 0\ \forall \mu_2 \in (0, \epsilon_2) :$$

$$c^* + \mu_1 \cdot d^* + \mu_2 \cdot e^* \in E_n \ .$$

Finally: M's behaviour using $c^* + \mu_1 \cdot d^* + \mu_2 \cdot e^*$ for μ_1, μ_2 as above can be simulated in $\mathrm{P}_{\mathbb{R}}^{\mathrm{rc}}$

Remarks and open problems

Remarks and open problems

Does Ladner's result hold in the full real BSS model?

Remarks and open problems

Does Ladner's result hold in the full real BSS model?

- limit arguments in semi-algebraic framework much more complicated

Remarks and open problems

Does Ladner's result hold in the full real BSS model?

- limit arguments in semi-algebraic framework much more complicated

Remarks and open problems

Does Ladner's result hold in the full real BSS model?

- limit arguments in semi-algebraic framework much more complicated
- if at all one can say something about a limit behaviour, a formula of type $\exists\forall\exists\forall$ likely cannot be evaluated in $\mathrm{P}_{\mathbb{R}}$

Remarks and open problems

Does Ladner's result hold in the full real BSS model?

- limit arguments in semi-algebraic framework much more complicated
- if at all one can say something about a limit behaviour, a formula of type $\exists\forall\exists\forall$ likely cannot be evaluated in $\mathrm{P}_{\mathbb{R}}$

Remarks and open problems

Does Ladner's result hold in the full real BSS model?

- limit arguments in semi-algebraic framework much more complicated
- if at all one can say something about a limit behaviour, a formula of type $\exists\forall\exists\forall$ likely cannot be evaluated in $\mathrm{P}_{\mathbb{R}}$ (no surprise since $\mathrm{P}_{\mathbb{R}} = \mathrm{P}_{\mathbb{R}}/\mathrm{const}$ unlikely!)
  similar quantifier structures studied by Bürgisser & Cucker

Remarks and open problems

Does Ladner's result hold in the full real BSS model?

- limit arguments in semi-algebraic framework much more complicated
- if at all one can say something about a limit behaviour, a formula of type $\exists\forall\exists\forall$ likely cannot be evaluated in $\mathrm{P}_{\mathbb{R}}$ (no surprise since $\mathrm{P}_{\mathbb{R}} = \mathrm{P}_{\mathbb{R}}/\mathrm{const}$ unlikely!)
  similar quantifier structures studied by Bürgisser & Cucker
- definition of class $\mathrm{P}_{\mathbb{R}}^{\mathrm{rc}}/\mathrm{const}$ allows certain degree of freedom; may be other definitions more helpful?

- what might be candidates for intermediate problems?

- what might be candidates for intermediate problems?

- what might be candidates for intermediate problems?

  Linear Programming: major open problem in optimization whether there are efficient algorithms for LP in algebraic model; the classical polynomial time algorithms like Interior Point, Ellipsoid method are not polynomial time in BSS

- what might be candidates for intermediate problems?

  Linear Programming: major open problem in optimization whether there are efficient algorithms for LP in algebraic model; the classical polynomial time algorithms like Interior Point, Ellipsoid method are not polynomial time in BSS

  Quadratic Programming: is NP-complete in Turing model, but likely not $\mathrm{NP}_{\mathbb{R}}$-complete because of discrete search space

- what might be candidates for intermediate problems?

  Linear Programming: major open problem in optimization whether there are efficient algorithms for LP in algebraic model; the classical polynomial time algorithms like Interior Point, Ellipsoid method are not polynomial time in BSS

  Quadratic Programming: is NP-complete in Turing model, but likely not $\mathrm{NP}_{\mathbb{R}}$-complete because of discrete search space

- Ladner type result for Valiant's complexity classes obtained by Bürgisser

4. Recursion theory over $\mathbb{R}$, Post's problem

Blum-Shub-Smale: Real Halting problem is BSS undecidable

$$\mathbb{H}_{\mathbb{R}} := \{\text{code of BSS machine } M \text{ that halts on empty input}\}$$

4. Recursion theory over $\mathbb{R}$, Post's problem

Blum-Shub-Smale: Real Halting problem is BSS undecidable

$$\mathbb{H}_{\mathbb{R}} := \{\text{code of BSS machine } M \text{ that halts on empty input}\}$$

further undecidable problems:

- $\mathbb{Q}$, i.e., given $x \in \mathbb{R}$, is x rational? Problem is semi-decidable: there is an algorithms which stops exactly for inputs from $\mathbb{Q}$;
- similarly for the algebraic real numbers $\mathbb{A}$:= set of real zeros of any polynomial $p \in \mathbb{Z}[x]$;
- Mandelbrot and certain Julia sets

Typical related questions:

- degrees of undecidability
- Post's problem: are there problems easier than $\mathbb{H}_{\mathbb{R}}$ yet undecidable?
- find other natural undecidable problems equivalent to $\mathbb{H}_{\mathbb{R}}$

Typical related questions:

- degrees of undecidability
- Post's problem: are there problems easier than $\mathbb{H}_{\mathbb{R}}$ yet undecidable?
- find other natural undecidable problems equivalent to $\mathbb{H}_{\mathbb{R}}$

Formalization of comparing problems via oracle machines:

A is Turing reducible to B iff A can be decided by a BSS machine that additionally has access to an oracle for membership in B.

A equivalent to B iff both are Turing reducible to each other

Real Post's problem: Are there problems Turing reducible to $\mathbb{H}_{\mathbb{R}}$ that are not Turing reducible from $\mathbb{H}_{\mathbb{R}}$ but yet undecidable?

Real Post's problem: Are there problems Turing reducible to $\mathbb{H}_{\mathbb{R}}$ that are not Turing reducible from $\mathbb{H}_{\mathbb{R}}$ but yet undecidable?

Turing setting: question posted in 1944 and solved 57/58 by Friedberg & Muchnik in the positive using diagonalization (finite injury priority technique);

however: no explicit problem with this property known so far

Real Post's problem: Are there problems Turing reducible to $\mathbb{H}_{\mathbb{R}}$ that are not Turing reducible from $\mathbb{H}_{\mathbb{R}}$ but yet undecidable?

Turing setting: question posted in 1944 and solved 57/58 by Friedberg & Muchnik in the positive using diagonalization (finite injury priority technique);

however: no explicit problem with this property known so far

Answer easier and more concrete over the reals!

Theorem (M. & Ziegler 2007)

The rational numbers $\mathbb{Q}$ are strictly easier than $\mathbb{H}_{\mathbb{R}}$.

Theorem (M. & Ziegler 2007)

The rational numbers $\mathbb{Q}$ are strictly easier than $\mathbb{H}_{\mathbb{R}}$.

Proof.

Show that $\mathbb{Q}$ is strictly easier than $\mathbb{A}$ by proving

i) $\mathbb{Q}$ is decidable with oracle for $\mathbb{A}$

ii) set $\mathbb{T} := \mathbb{R} \setminus \mathbb{A}$ of transcendent reals is not semi-decidable even with oracle for $\mathbb{Q}$

Theorem (M. & Ziegler 2007)

The rational numbers $\mathbb{Q}$ are strictly easier than $\mathbb{H}_{\mathbb{R}}$.

Proof.

Show that $\mathbb{Q}$ is strictly easier than $\mathbb{A}$ by proving

i) $\mathbb{Q}$ is decidable with oracle for $\mathbb{A}$

ii) set $\mathbb{T} := \mathbb{R} \setminus \mathbb{A}$ of transcendent reals is not semi-decidable even with oracle for $\mathbb{Q}$

implies claim because

Theorem (M. & Ziegler 2007)

The rational numbers $\mathbb{Q}$ are strictly easier than $\mathbb{H}_{\mathbb{R}}$.

Proof.

Show that $\mathbb{Q}$ is strictly easier than $\mathbb{A}$ by proving

i) $\mathbb{Q}$ is decidable with oracle for $\mathbb{A}$

ii) set $\mathbb{T} := \mathbb{R} \setminus \mathbb{A}$ of transcendent reals is not semi-decidable even with oracle for $\mathbb{Q}$

implies claim because if $\mathbb{H}_{\mathbb{R}}$ was decidable with oracle for $\mathbb{Q}$, then $\mathbb{A}$ as well: as semi-decidable problem it can be decided with oracle $\mathbb{H}_{\mathbb{R}}$;

Theorem (M. & Ziegler 2007)

The rational numbers $\mathbb{Q}$ are strictly easier than $\mathbb{H}_{\mathbb{R}}$.

Proof.

Show that $\mathbb{Q}$ is strictly easier than $\mathbb{A}$ by proving

i) $\mathbb{Q}$ is decidable with oracle for $\mathbb{A}$

ii) set $\mathbb{T} := \mathbb{R} \setminus \mathbb{A}$ of transcendent reals is not semi-decidable even with oracle for $\mathbb{Q}$

implies claim because if $\mathbb{H}_{\mathbb{R}}$ was decidable with oracle for $\mathbb{Q}$, then $\mathbb{A}$ as well: as semi-decidable problem it can be decided with oracle $\mathbb{H}_{\mathbb{R}}$; contradiction to ii) because then also $\mathbb{T}$ is decidable with oracle $\mathbb{Q}$.

Proof (cntd.)

ad i) $\mathbb{Q}$ is decidable with oracle for $\mathbb{A}$:

function *deg* : $\mathbb{A} \mapsto \mathbb{N}_0$, $deg(a) =$ degree of algebraic number a is BSS-computable:

Proof (cntd.)

ad i) $\mathbb{Q}$ is decidable with oracle for $\mathbb{A}$:

function *deg* : $\mathbb{A} \mapsto \mathbb{N}_0, deg(a) =$ degree of algebraic number a is BSS-computable:

given $a \in \mathbb{A}$, try all irreducible $p \in \mathbb{Z}[x]$ for $p(a) = 0$? If p is found its degree gives $deg(a)$; has to happen because $a \in \mathbb{A}$ is assumed!

Note: irreducibility over $\mathbb{Z}[x]$ in NP (Cantor 1981), thus also decidable in BSS model

Proof (cntd.)

ad ii) $\mathbb{T}$ not semi-decidable with oracle for $\mathbb{Q}$:

Proof (cntd.)

ad ii) $\mathbb{T}$ not semi-decidable with oracle for $\mathbb{Q}$:

assume otherwise; oracle computation for input $x \in \mathbb{R}$ can branch on sign and (ir-)rationality of intermediate results; such results have form $f(x)$ for an $f \in \mathbb{R}(x)$;

branches: $f(x) < 0?, f(x) = 0?, f(x) > 0?, f(x) \in \mathbb{Q}?, f(x) \notin \mathbb{Q}$

Proof (cntd.)

ad ii) $\mathbb{T}$ not semi-decidable with oracle for $\mathbb{Q}$:

assume otherwise; oracle computation for input $x \in \mathbb{R}$ can branch on sign and (ir-)rationality of intermediate results; such results have form $f(x)$ for an $f \in \mathbb{R}(x)$;

branches: $f(x) < 0?, f(x) = 0?, f(x) > 0?, f(x) \in \mathbb{Q}?, f(x) \notin \mathbb{Q}$

oracle computation corresponds to (possibly infinite) computation tree, in particular all $x \in \mathbb{T}$ end at a leaf;

since tree has countably many paths only there exists a finite computation path φ leading to a leaf that branches uncountably many inputs from $\mathbb{T}$: denote this set by U

Proof (cntd.)

Let $\{f_u(x)\}_u$ denote the finitely many intermediate results computed along φ that enter a branch instruction;

Proof (cntd.)

Let $\{f_u(x)\}_u$ denote the finitely many intermediate results computed along φ that enter a branch instruction;

all f_u are rational non-constant functions; the corresponding tests have special outcome:

- there is no result $f_u(x) = 0$ for $x \in U$ along ϕ; otherwise test branches only finitely many points;

Proof (cntd.)

Let $\{f_u(x)\}_u$ denote the finitely many intermediate results computed along φ that enter a branch instruction;

all f_u are rational non-constant functions; the corresponding tests have special outcome:

- there is no result $f_u(x) = 0$ for $x \in U$ along ϕ; otherwise test branches only finitely many points;
- there is no result $f_u(x) \in \mathbb{Q}$ along φ; no non-constant analytical f_u maps an uncountable set into $\mathbb{Q}$

Proof (cntd.)

- Theorem: For each rational f_u there is an integer D such that $f(a) \in \mathbb{Q}$ only for algebraic a of degree at most D

Continuity of the finitely many f_u and the previous theorem imply that all $x \in \mathbb{A}$ of sufficiently high degree are branched along φ as well; thus the algorithm errs on those inputs!

Proof (cntd.)

- Theorem: For each rational f_u there is an integer D such that $f(a) \in \mathbb{Q}$ only for algebraic a of degree at most D

Continuity of the finitely many f_u and the previous theorem imply that all $x \in \mathbb{A}$ of sufficiently high degree are branched along φ as well; thus the algorithm errs on those inputs!

There are concrete problems of the same degree as the real Halting problem:

Word problem for certain groups (M. & Ziegler 2009)

Word Problem for Groups I

Consider product bab^2ab^2aba in free semi-group $\langle\{a, b\}\rangle$;

subject to

- rule $ab = 1$ it can be simplified to b^2 but not to 1
- using additional rules $a^4 = a^2$ it can be simplified to 1

Word Problem for Groups I

Consider product bab^2ab^2aba in free semi-group $\langle\{a,b\}\rangle$;

subject to

- rule $ab = 1$ it can be simplified to b^2 but not to 1
- using additional rules $a^4 = a^2$ it can be simplified to 1

Fix set X and set R of equations over $\langle X\rangle = (X \cup X^{-1})^*$.

Word problem for $\langle X\rangle$: Given a formal product

$w := x_1^{\pm 1} x_2^{\pm 1 \dots} x_n^{\pm 1}, x_i \in X$, does it hold subject to R that $w = 1$?

Boone '58, Novikov '59: There exist finite X, R such that the related word problem is equivalent to discrete Halting problem.

Word Problem for Groups II

Now set $X \subset \mathbb{R}^*$ of real generators, R rules on $\langle X \rangle$;

word problem as before, but suitable for BSS setting

Example

$X := \{x_r | r \in \mathbb{R}\}; R := \{x_{nr} = x_r, x_{r+k} = x_r | r \in \mathbb{R}, n \in \mathbb{N}, k \in \mathbb{Z}\}$

X, R are BSS decidable and $x_r = 1 \Leftrightarrow$

Word Problem for Groups II

Now set $X \subset \mathbb{R}^*$ of real generators, R rules on $\langle X \rangle$;

word problem as before, but suitable for BSS setting

Example

$X := \{x_r | r \in \mathbb{R}\}; R := \{x_{nr} = x_r, x_{r+k} = x_r | r \in \mathbb{R}, n \in \mathbb{N}, k \in \mathbb{Z}\}$

X, R are BSS decidable and $x_r = 1 \Leftrightarrow r \in \mathbb{Q}$

Word Problem for Groups II

Now set $X \subset \mathbb{R}^*$ of real generators, R rules on $\langle X \rangle$;

word problem as before, but suitable for BSS setting

Example

$X := \{x_r | r \in \mathbb{R}\}; R := \{x_{nr} = x_r, x_{r+k} = x_r | r \in \mathbb{R}, n \in \mathbb{N}, k \in \mathbb{Z}\}$

X, R are BSS decidable and $x_r = 1 \Leftrightarrow r \in \mathbb{Q}$

Thus this world problem is undecidable, but easier than $\mathbb{H}_{\mathbb{R}}$.

Theorem (M. & Ziegler 2009)

There are BSS decidable sets $X \subset \mathbb{R}^N$, $R \subset \mathbb{R}^*$ such that the resulting word problem is equivalent to $\mathbb{H}_{\mathbb{R}}$.

Proof.

Lot of combinatorial group theory: Nielsen reduction, HNN extensions, Britton's Lemma, amalgamation, ...

Theorem (M. & Ziegler 2009)

There are BSS decidable sets $X \subset \mathbb{R}^N$, $R \subset \mathbb{R}^*$ such that the resulting word problem is equivalent to $\mathbb{H}_{\mathbb{R}}$.

Proof.

Lot of combinatorial group theory: Nielsen reduction, HNN extensions, Britton's Lemma, amalgamation, ...

Reals enter as index set for set of generators; no particular influence of semi-algebraic geometry; word problem is located in computational group theory and thus presents new kind of complete problem in BSS recursion theory.

Further research questions:

- power of other undecidable problems like Mandelbrot set?
- use of machine constants: what power does one gain by using more machine constants?
- find word problems representing real number complexity classes like $\mathrm{NP}_{\mathbb{R}}$ or $\mathrm{P}_{\mathbb{R}}$
- Bounded query computations: how many queries to an oracle B are needed to compute characteristic function χ_n^A for A^n on $(\mathbb{R}^*)^n$?

  Example: For $A = B = \mathbb{H}_{\mathbb{R}}$ $\log n$ queries are sufficient.

Summary for today:

Many problems important in Turing framework have natural formulation as well in real / complex number models

Summary for today:

Many problems important in Turing framework have natural formulation as well in real / complex number models

all kinds of aspects can occur as to whether results can be transferred between models, such as:

- results by easy reasoning hold as well

Summary for today:

Many problems important in Turing framework have natural formulation as well in real / complex number models

all kinds of aspects can occur as to whether results can be transferred between models, such as:

- results by easy reasoning hold as well
- results hold, but need much deeper arguments

Summary for today:

Many problems important in Turing framework have natural formulation as well in real / complex number models

all kinds of aspects can occur as to whether results can be transferred between models, such as:

- results by easy reasoning hold as well
- results hold, but need much deeper arguments
- ... or by easier arguments

Summary for today:

Many problems important in Turing framework have natural formulation as well in real / complex number models

all kinds of aspects can occur as to whether results can be transferred between models, such as:

- results by easy reasoning hold as well
- results hold, but need much deeper arguments
- ... or by easier arguments
- new questions arise due to different mathematical structure

Summary for today:

Many problems important in Turing framework have natural formulation as well in real / complex number models

all kinds of aspects can occur as to whether results can be transferred between models, such as:

- results by easy reasoning hold as well
- results hold, but need much deeper arguments
- ... or by easier arguments
- new questions arise due to different mathematical structure
- real and complex classes related to major open classical problems

Summary for today:

Many problems important in Turing framework have natural formulation as well in real / complex number models

all kinds of aspects can occur as to whether results can be transferred between models, such as:

- results by easy reasoning hold as well
- results hold, but need much deeper arguments
- ... or by easier arguments
- new questions arise due to different mathematical structure
- real and complex classes related to major open classical problems