Real Interactive Proofs for VPSPACE

Klaus Meer

Brandenburgische Technische Universität, Cottbus-Senftenberg, Germany

Colloquium Logicum
Hamburg, September 2016

joint work with M. Baartse

1. Introduction

Blum-Shub-Smale model of computability and complexity over $\mathbb{R}$:

Algorithms allow as basic steps arithmetic operations $+, -, \cdot$ as well as test operation '$x \geq 0$?'

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

Definition of complexity classes $\mathrm{P}_{\mathbb{R}}, \mathrm{NP}_{\mathbb{R}}, \mathrm{PAR}_{\mathbb{R}}, \mathrm{PAT}_{\mathbb{R}}$,..., completeness notions for those classes, real version of P versus NP question etc.

Inspiring source of interesting questions in BSS model:

which form do classical theorems (Turing model) take?

Inspiring source of interesting questions in BSS model: which form do classical theorems (Turing model) take?

Examples:

- decidability of problems in $\mathrm{NP}_{\mathbb{R}}$ (Grigoriev, Vorobjov, Heintz, Renegar ...)
- transfer theorems (Shub&Smale, Koiran,...)
- complexity separations: $\mathrm{P}_{\mathbb{R}} \neq \mathrm{NC}_{\mathbb{R}}$ (Cucker)
- real complexity of Boolean languages (Bürgisser, Cucker, Grigoriev, Koiran,...)
- Toda's theorem (Basu & Zell)
- real PCP theorem (Baartse & M.)
- ...

Here: Interactive Proofs and Shamir's theorem

Theorem (Shamir 1992)

$\mathrm{IP} = \mathrm{PSPACE} \;\; (= \mathrm{PAR} = \mathrm{PAT})$

Here: Interactive Proofs and Shamir's theorem

Theorem (Shamir 1992)

$\mathrm{IP} = \mathrm{PSPACE} \;\; (= \mathrm{PAR} = \mathrm{PAT})$

Problem over $\mathbb{R}$: space resources alone meaningless, real analogues $\mathrm{PAR}_{\mathbb{R}}$ and $\mathrm{PAT}_{\mathbb{R}}$ differ:

Theorem (Cucker 1994)

$\mathrm{PAR}_{\mathbb{R}} \subsetneq \mathrm{PAT}_{\mathbb{R}}$

Questions:

- Is a real version $\mathrm{IP}_{\mathbb{R}}$ still captured by one of the two classes?
- Or by something different?
- Upper bounds for $\mathrm{IP}_{\mathbb{R}}$?
- Lower bounds for $\mathrm{IP}_{\mathbb{R}}$?
- How far does Shamir's discrete technique lead?

2. Interactive proofs over $\mathbb{R}$: Class $\mathrm{IP}_{\mathbb{R}}$

Prover P: BSS machine of unlimited power
Verifier V: randomized polynomial time BSS algorithm; V generates sequence of random bits $r = (r_1, r_2, \ldots)$

2. Interactive proofs over $\mathbb{R}$: Class $\mathrm{IP}_{\mathbb{R}}$

Prover P: BSS machine of unlimited power
Verifier V: randomized polynomial time BSS algorithm; V generates sequence of random bits $r = (r_1, r_2, \ldots)$
Computation proceeds as follows:

- on input $x \in \mathbb{R}^n$ and (some of) the random bits V computes a real $V(x, r) =: w_1 \in \mathbb{R}$ and sends it to P;

2. Interactive proofs over $\mathbb{R}$: Class $\mathrm{IP}_{\mathbb{R}}$

Prover P: BSS machine of unlimited power
Verifier V: randomized polynomial time BSS algorithm; V generates sequence of random bits $r = (r_1, r_2, \ldots)$
Computation proceeds as follows:

- on input $x \in \mathbb{R}^n$ and (some of) the random bits V computes a real $V(x, r) =: w_1 \in \mathbb{R}$ and sends it to P;
- P sends a real $P(x, w_1) =: p_1 \in \mathbb{R}$ back to V;

2. Interactive proofs over $\mathbb{R}$: Class $\mathrm{IP}_{\mathbb{R}}$

Prover P: BSS machine of unlimited power
Verifier V: randomized polynomial time BSS algorithm; V generates sequence of random bits $r = (r_1, r_2, \ldots)$
Computation proceeds as follows:

- on input $x \in \mathbb{R}^n$ and (some of) the random bits V computes a real $V(x, r) =: w_1 \in \mathbb{R}$ and sends it to P;
- P sends a real $P(x, w_1) =: p_1 \in \mathbb{R}$ back to V;
- using information sent forth and back after i rounds V computes real $V(x, r, w_1, p_1, \ldots, p_i) =: w_{i+1}$ and sends it to P; P computes a real p_{i+1} and sends it to V;

2. Interactive proofs over $\mathbb{R}$: Class $\mathrm{IP}_{\mathbb{R}}$

Prover P: BSS machine of unlimited power
Verifier V: randomized polynomial time BSS algorithm; V generates sequence of random bits $r = (r_1, r_2, \ldots)$
Computation proceeds as follows:

- on input $x \in \mathbb{R}^n$ and (some of) the random bits V computes a real $V(x, r) =: w_1 \in \mathbb{R}$ and sends it to P;
- P sends a real $P(x, w_1) =: p_1 \in \mathbb{R}$ back to V;
- using information sent forth and back after i rounds V computes real $V(x, r, w_1, p_1, \ldots, p_i) =: w_{i+1}$ and sends it to P; P computes a real p_{i+1} and sends it to V;
- communication halts after a polynomial number m of rounds. Final result $V(x, r, w_1, \ldots, p_{m-1}) =: w_m \in \{0, 1\}$ reject / accept

2. Interactive proofs over $\mathbb{R}$: Class $\mathrm{IP}_{\mathbb{R}}$

Prover P: BSS machine of unlimited power
Verifier V: randomized polynomial time BSS algorithm; V generates sequence of random bits $r = (r_1, r_2, \ldots)$
Computation proceeds as follows:

- on input $x \in \mathbb{R}^n$ and (some of) the random bits V computes a real $V(x, r) =: w_1 \in \mathbb{R}$ and sends it to P;
- P sends a real $P(x, w_1) =: p_1 \in \mathbb{R}$ back to V;
- using information sent forth and back after i rounds V computes real $V(x, r, w_1, p_1, \ldots, p_i) =: w_{i+1}$ and sends it to P; P computes a real p_{i+1} and sends it to V;
- communication halts after a polynomial number m of rounds. Final result $V(x, r, w_1, \ldots, p_{m-1}) =: w_m \in \{0, 1\}$ reject / accept

2. Interactive proofs over $\mathbb{R}$: Class $\mathrm{IP}_{\mathbb{R}}$

Prover P: BSS machine of unlimited power
Verifier V: randomized polynomial time BSS algorithm; V generates sequence of random bits $r = (r_1, r_2, \ldots)$
Computation proceeds as follows:

- on input $x \in \mathbb{R}^n$ and (some of) the random bits V computes a real $V(x, r) =: w_1 \in \mathbb{R}$ and sends it to P;
- P sends a real $P(x, w_1) =: p_1 \in \mathbb{R}$ back to V;
- using information sent forth and back after i rounds V computes real $V(x, r, w_1, p_1, \ldots, p_i) =: w_{i+1}$ and sends it to P; P computes a real p_{i+1} and sends it to V;
- communication halts after a polynomial number m of rounds. Final result $V(x, r, w_1, \ldots, p_{m-1}) =: w_m \in \{0, 1\}$ reject / accept

$(P, V)(x, r)$ = result of interaction on x using r

Definition

$L \in \mathrm{IP}_{\mathbb{R}}$ iff there exists a polynomial time randomized verifier V such that

i) if $x \in L$ there exists a prover P such that $\Pr_{r \in \{0,1\}^*} \{(P,V)(x,r) = 1\} = 1$ and

ii) if $x \notin L$, then for all provers P it is $\Pr_{r \in \{0,1\}^*} \{(P,V)(x,r) = 1\} \leq \frac{1}{4}$.

Definition

$L \in \mathrm{IP}_{\mathbb{R}}$ iff there exists a polynomial time randomized verifier V such that

i) if $x \in L$ there exists a prover P such that
$$\Pr_{r \in \{0,1\}^*} \{(P,V)(x,r) = 1\} = 1 \text{ and}$$

ii) if $x \notin L$, then for all provers P it is
$$\Pr_{r \in \{0,1\}^*} \{(P,V)(x,r) = 1\} \leq \frac{1}{4}.$$

Remark: Class $\mathrm{IP}_{\mathbb{R}}$ remains the same when using public coins and/or two-sided error.

Previous results: Ivanov & de Rougemont study interactive proofs in additive BSS model exchanging bits and show
$\mathrm{PAR}_{\mathbb{R},+} = \mathrm{BIP}_{\mathbb{R},+}$

Important for us: they design a problem outside $\mathrm{PAR}_{\mathbb{R}}$ that has an additive interactive proof in which reals are exchanged (problem considered for Cucker's 1994 result)

Previous results: Ivanov & de Rougemont study interactive proofs in additive BSS model exchanging bits and show
$\mathrm{PAR}_{\mathbb{R},+} = \mathrm{BIP}_{\mathbb{R},+}$

Important for us: they design a problem outside $\mathrm{PAR}_{\mathbb{R}}$ that has an additive interactive proof in which reals are exchanged (problem considered for Cucker's 1994 result)

Consequence: $\mathrm{PAR}_{\mathbb{R}} \neq \mathrm{IP}_{\mathbb{R}}$

But: No significant upper or lower bounds for (full) $\mathrm{IP}_{\mathbb{R}}$ known

3. Upper bound: The class $\mathrm{MA}\exists\mathbb{R}$ of mixed alternation

Description of interaction protocols roughly as follows:

- computation for exponentially many random strings generated by V can be covered in parallel;
- search an optimal prover: look for optimal real answers the prover sends to V in order to imply maximal number of random strings leading to accepting protocol

3. Upper bound: The class $\mathrm{MA}\exists\mathbb{R}$ of mixed alternation

Description of interaction protocols roughly as follows:

- computation for exponentially many random strings generated by V can be covered in parallel;
- search an optimal prover: look for optimal real answers the prover sends to V in order to imply maximal number of random strings leading to accepting protocol

Second item leads to additional existential real quantifiers on top of parallel computation

Suitable complexity class introduced by Briquel & Cucker: $\mathrm{MA}\exists\mathbb{R}$

Definition (Mixed alternation)

$A \in \mathrm{MA}\exists\mathbb{R}$ iff there exists $L \in \mathrm{P}_{\mathbb{R}}$ and polynomial p such that $x \in A$ if and only if the following formula holds:

$$\forall_B z_1 \exists_{\mathbb{R}} y_1 \ldots \forall_B z_{p(|x|)} \exists_{\mathbb{R}} y_{p(|x|)} (x, y, z) \in L \ .$$

The subscripts $B, \mathbb{R}$ for the quantifiers indicate whether a quantified variable ranges over $B := \{0, 1\}$ or $\mathbb{R}$, respectively

i.e., polynomially alternating formula with arbitrary Boolean and existential real quantifiers

Suitable complexity class introduced by Briquel & Cucker: $\mathrm{MA}\exists\mathbb{R}$

Definition (Mixed alternation)

$A \in \mathrm{MA}\exists\mathbb{R}$ iff there exists $L \in \mathrm{P}_{\mathbb{R}}$ and polynomial p such that $x \in A$ if and only if the following formula holds:

$$\forall_B z_1 \exists_{\mathbb{R}} y_1 \ldots \forall_B z_{p(|x|)} \exists_{\mathbb{R}} y_{p(|x|)} (x, y, z) \in L \; .$$

The subscripts $B, \mathbb{R}$ for the quantifiers indicate whether a quantified variable ranges over $B := \{0, 1\}$ or $\mathbb{R}$, respectively

i.e., polynomially alternating formula with arbitrary Boolean and existential real quantifiers

Cucker & Briquel: $\mathrm{PAR}_{\mathbb{R}} \subsetneq \mathrm{MA}\exists\mathbb{R} \subseteq \mathrm{PAT}_{\mathbb{R}}$

Theorem (Baartse & M. 2015)

It holds $\mathrm{IP}_{\mathbb{R}} \subseteq \mathrm{MA}\exists\mathbb{R}$

Theorem (Baartse & M. 2015)

It holds $\mathrm{IP}_{\mathbb{R}} \subseteq \mathrm{MA}\exists\mathbb{R}$

Proof formalizes above idea of describing an optimal protocol

4. Lower bounds: MFCS contribution

Upper bound shows: $\mathrm{IP}_{\mathbb{R}} \subseteq \mathrm{MA}\exists\mathbb{R} \subseteq \mathrm{PAT}_{\mathbb{R}}$; the latter inclusion is conjectured to be strict, thus $\mathrm{IP}_{\mathbb{R}}$ likely strictly included in $\mathrm{PAT}_{\mathbb{R}}$;

result by Ivanov and de Rougemont shows: $\mathrm{IP}_{\mathbb{R}} \neq \mathrm{PAR}_{\mathbb{R}}$

Can we design interactive protocols for interesting real complexity classes?
How far does Shamir's discrete technique lead?

Recall Shamir's technique to design IP for QBF:

- arithmetization of formula gives short algebraic expression replacing quantifiers by operators $\sum\limits_{x_i \in \{0,1\}} x_i$, $\prod\limits_{x_i \in \{0,1\}} x_i$ ranging over $\{0,1\}$; explicit expression has exponentially many terms;
- recursively attach canonical univariate polynomials of polynomial degree to expression by eliminating leftmost $\sum\limits_{x_i=0}^{1}$ or $\prod\limits_{x_i=0}^{1}$
- verify value of those polynomials in random points interactively

Clear: Arithmetization breaks down when quantifiers range over $\mathbb{R}$
Question: Can Shamir's technique anyway be used?

Definition (Koiran & Perifel)

Family $\{f_n\}_{n\in\mathbb{N}}$ of real polynomials is in *UniformVPSPACE* iff there exists a polynomial p such that

i) each f_n depends on $p(n)$ variables

ii) total degree of f_n bounded by $2^{p(n)}$;

iii) coefficients of f_n integers of bit size $\leq 2^{p(n)} - 1$;

iv) coefficient function a is PSPACE computable; $a(n, \alpha, i) \in \{0, 1\}$ gives the i-th bit of the coefficient of monomial x^α in f_n (and $a(n, \alpha, 0)$ gives the sign)

$$f_n(x_1, \ldots, x_{u(n)}) = \sum_\alpha \left[(-1)^{a(n,\alpha,0)} \left(\sum_{i=1}^{2^{p(n)}} 2^{i-1} a(n, \alpha, i) \right) x^\alpha \right].$$

Koiran & Perifel:

- class *UniformVPSPACE* generalizes VNP
- all problems in $\mathrm{PAR}_{\mathbb{R}}$ can be decided by a polynomial time BSS oracle algorithm using an oracle for evaluating functions of a family $\{f_n\}_n \in \mathit{UniformVPSPACE}$

Theorem

UniformVPSPACE $\subseteq \mathrm{IP}_{\mathbb{R}}$ *in the following sense: For* $\{f_n\}_n \in$ *UniformVPSPACE there exists an interactive protocol for the language* $\{(n,x,y) \in \mathbb{N} \times \mathbb{R}^{p(n)} \times \mathbb{R} \mid f_n(x) = y\}$.

Proof.

Functions in *UniformVPSPACE* can be described via a discrete construction pattern resembling structure of Shamir's arithmetization of QBF

Proof (cntd.)

Binary polynomial formula bpf over reals is a formula p built in finitely many steps according to rules:

i) $p = 1$ and $p = x_i$ for $i = 1, 2, \ldots$ are bpf;

ii) if p_1, p_2 are binary polynomial formulas, then so are $p_1 + p_2, p_1 - p_2, p_1 \cdot p_2$;

iii) if p is bpf depending freely on x_i, then both $\sum_{x_i \in \{0,1\}} p(\ldots, x_i, \ldots)$ and $\prod_{x_i \in \{0,1\}} p(\ldots, x_i, \ldots)$ are bpf

Size of p: # construction steps

pbf canonically represents a real polynomial function in its free variables

Proof (cntd.)

We need following relation of bpf to *UniformVPSPACE*:

Theorem (similar results by Poizat, Malod)

Let $\{f_n\}_n$ be a family of polynomial functions. Then $\{f_n\}_n \in$ UniformVPSPACE if and only if there exists a polynomial time Turing algorithm which on input $n \in \mathbb{N}$ (in unary) computes a binary polynomial formula p_n which represents f_n.

Proof constructs pdf for all parts of the representation

$$f_n(x_1, \ldots, x_{u(n)}) = \sum_{\alpha} \left[(-1)^{a(n,\alpha,0)} \left(\sum_{i=1}^{2^{p(n)}} 2^{i-1} a(n,\alpha,i) \right) x^{\alpha} \right].$$

Proof (cntd.)

interactive protocol for verifying correct evaluation of f_n:

- construct corresponding pbf for f_n and
- apply Shamir's technique to the latter

Proof (cntd.)

interactive protocol for verifying correct evaluation of f_n:

- construct corresponding pbf for f_n and
- apply Shamir's technique to the latter

Using result by Koiran & Perifel this implies lower bound

Theorem

$\mathrm{PAR}_{\mathbb{R}} \subsetneq \mathrm{IP}_{\mathbb{R}} \subseteq \mathrm{MA}\exists\mathbb{R}$

Open questions

- How large is the class $\mathrm{P}_{\mathbb{R}}^{\mathit{UniformVPSPACE}}$? How far does approach with oracle computations lead?
- Is $\mathrm{IP}_{\mathbb{R}}$ closed under complementation?
- Possible characterization of $\mathrm{IP}_{\mathbb{R}}$: class $\mathrm{PSPACE}_{\mathbb{R}}$ of problems decidable in polynomial space by $\mathsf{EXPTIME}_{\mathbb{R}}$ algorithm
known: $\mathrm{PAR}_{\mathbb{R}} \subsetneq \mathsf{PSPACE}_{\mathbb{R}} \subseteq \mathrm{MA}\exists\mathbb{R}$

Definition (Real Parallel Time $\mathrm{PAR}_\mathbb{R}$)

A problem $L \subseteq \mathbb{R}^\infty := \bigsqcup_{i \geq 1} \mathbb{R}^i$ belongs to class $\mathrm{PAR}_\mathbb{R}$ iff there exists a family $\{C_n\}_{n\in\mathbb{N}}$ of algebraic circuits of depth polynomially bounded in n, a constant $s \in \mathbb{N}$, and a vector $c \in \mathbb{R}^s$ of real constants such that

i) each C_n has $n + s$ input nodes;
ii) for all $n \in \mathbb{N}$ the circuit C_n computes the characteristic function of $L \cap \mathbb{R}^n$, when the last s input nodes are assigned the constant values from c, i.e., $x \in L \cap \mathbb{R}^n \Leftrightarrow C_n(x, c) = 1$;
iii) the family $\{C_n\}_n$ is PSPACE uniform, i.e., there is a Turing machine working in polynomial space which for each $n \in \mathbb{N}$ computes a description of C_n.

If no constant vector c is involved we obtain the constant free version of $\mathrm{PAR}_\mathbb{R}$ denoted by $\mathrm{PAR}^0_\mathbb{R}$.

Definition (Polynomial Alternating Time $\mathrm{PAT}_{\mathbb{R}}$)

$A \in \mathrm{PAT}_{\mathbb{R}}$ iff there exists $L \in \mathrm{P}_{\mathbb{R}}$ and polynomial p such that $x \in A$ if and only if the following formula holds:

$$\forall_{\mathbb{R}} z_1 \exists_{\mathbb{R}} y_1 \ldots \forall_{\mathbb{R}} z_{p(|x|)} \exists_{\mathbb{R}} y_{p(|x|)} (x, y, z) \in L \ .$$

The subscript $\mathbb{R}$ again indicates quantifiers ranging over $\mathbb{R}$.