Some Results on Interactive Proofs for Real Computations

Martijn Baartse and Klaus Meer(✉)

Computer Science Institute, BTU Cottbus-Senftenberg,
Platz der Deutschen Einheit 1, 03046 Cottbus, Germany
baartse@tu-cottbus.de, meer@b-tu.de

Abstract. We study interactive proofs in the framework of real number complexity theory as introduced by Blum, Shub, and Smale. Shamir's famous result characterizes the class IP as PSPACE or, equivalently, as PAT and PAR in the Turing model. Since space resources alone are known not to make much sense in real number computations the question arises whether IP can be similarly characterized by one of the latter classes. Ivanov and de Rougemont [9] started this line of research showing that an analogue of Shamir's result holds in the additive Blum-Shub-Smale model of computation when only Boolean messages can be exchanged. Here, we introduce interactive proofs in the full BSS model. As main result we prove an upper bound for the class $\mathrm{IP}_{\mathbb{R}}$. It gives rise to the conjecture that a characterization of $\mathrm{IP}_{\mathbb{R}}$ will not be given via one of the real complexity classes $\mathrm{PAR}_{\mathbb{R}}$ or $\mathrm{PAT}_{\mathbb{R}}$. We report on ongoing approaches to prove as well interesting lower bounds for $\mathrm{IP}_{\mathbb{R}}$.

1 Introduction

One inspiring source of research problems to deal with in the framework of real number computations [4] is the question, whether important results from Turing complexity theory hold as well over the reals as underlying structure. And in case they do what does it need to prove them. Beside the importance for the respective computational model this might as well shed light on a better understanding what is the intrinsic reason for a result to hold. Examples are provided by the huge amount of research on quantifier elimination algorithms yielding decidability of all problems in $\mathrm{NP}_{\mathbb{R}}$, a recent analogue of Toda's theorem [3], or the proof of a real version of the classical PCP theorem [2], to mention only a few.

Along the same lines in [9] the authors introduced interactive proofs in the framework of real number complexity theory. More precisely, they considered the additive version of the BSS model, see [4] and interaction is restricted to exchange boolean messages only. In this setting, IP again can be characterized via parallel polynomial time.

K. Meer—Both authors were supported under projects ME 1424/7-1 and ME 1424/7-2 by the Deutsche Forschungsgemeinschaft DFG. We gratefully acknowledge the support.

A. Beckmann et al. (Eds.): CiE 2015, LNCS 9136, pp. 107–116, 2015.
DOI: 10.1007/978-3-319-20028-6_11

It is thus natural to extend the definition of interactive proofs to the full BSS model by allowing real messages to be exchanged and the verifier to use multiplications as well. By a folklore result in real number complexity [11] each real decision problem can be decided in linear space. Thus the class $\mathrm{IP}_{\mathbb{R}}$ of real decision problems acceptable by such an interactive protocol cannot meaningfully be characterized using space resources alone. As with other real complexity classes that classically correspond to space classes the question then is whether and which other characterizations hold. Note that the two classes PAR and PAT of problems decidable in parallel and in alternating polynomial time, respectively, equal PSPACE in the Turing model. However, for their real counterparts $\mathrm{PAR}_{\mathbb{R}}, \mathrm{PAT}_{\mathbb{R}}$ it is known that the first is a proper subset of the latter [7]. So the question is whether one of them (and which) equals $\mathrm{IP}_{\mathbb{R}}$?

It has been shown in [9] that $\mathrm{PAR}_{\mathbb{R}}$ is different from $\mathrm{IP}_{\mathbb{R}}$; there are problems in $\mathrm{IP}_{\mathbb{R}}$ that cannot be solved in parallel polynomial time. A reason for this is that $\mathrm{PAR}_{\mathbb{R}}$ is too weak to capture certain real quantifier elimination tasks. One might then expect that the larger class $\mathrm{PAT}_{\mathbb{R}}$ is the correct one to capture $\mathrm{IP}_{\mathbb{R}}$. However, as our main result shows, this seems unlikely. We establish as an upper bound for $\mathrm{IP}_{\mathbb{R}}$ the class $\mathrm{MA}\exists_{\mathbb{R}}$. It was introduced in [8], is strictly larger than $\mathrm{PAR}_{\mathbb{R}}$ but conjectured to be strictly included in $\mathrm{PAT}_{\mathbb{R}}$. After introducing interactive protocols and the real complexity classes important for this paper we prove the upper bound result in Sect. 3. The remaining part of the paper then discusses approaches to achieve good lower bounds for $\mathrm{IP}_{\mathbb{R}}$ as well. This, however, seems to be much harder and we only report on some problems for which Shamir's classical technique can be extended to design real protocols as well. We try to substantiate why the lower bound problem in the full model seems harder by re-analyzing the result of Ivanov and de Rougemont using counting problems. This viewpoint leads to a couple of interesting open problems which seem important on the way to a full real analogue of Shamir's theorem.

2 Basic Notions

As underlying algorithm model we work in the Blum-Shub-Smale BSS model over $\mathbb{R}$ [4]. Decision problems considered in this model are subsets of $\mathbb{R}^\infty := \bigsqcup_{i\geq 1} \mathbb{R}^i$. The model allows to perform the basic arithmetic operations $+, -, \times$ and test instructions of the form 'is $x \geq 0$?' at unit cost. Below, in addition we allow both the verifier and the prover to exchange real numbers at unit cost.

The prover P is a BSS machine unlimited in computational power. The verifier V is a randomized polynomial time algorithm. It is important to point out that randomization (still) is discrete, i.e., V generates a sequence of random bits $r = (r_1, r_2, \ldots)$ during its computation. Now, the computation proceeds as follows:

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

– in general, if after i rounds of communication $(w_1, p_1, w_2, \ldots, p_i)$ denotes the information sent forth and back, in round $i+1$ V computes a real $V(x, r, w_1, p_1, \ldots, p_i) =: w_{i+1}$ and sends it to P; P then computes a real $P(x, r, w_1, p_1, \ldots, p_i, w_{i+1}) =: p_{i+1}$ and sends it to V;
– the communication halts after a polynomial number $m = poly(|x|)$ of rounds. Then V computes its final result $V(x, r, w_1, \ldots, p_{m-1}) =: w_m \in \{0, 1\}$ representing its decision to reject or accept the input, respectively.

We denote the result of an interaction between V and P on input x and using r as random string by $(P, V)(x, r)$. All computations by V have to be finished in (real) polynomial time; thus, in particular the number of random bits generated as well as the number of rounds is polynomially bounded in the algebraic size $|x|$ of x.

Definition 1. *(a) A language $L \subseteq \mathbb{R}^\infty$ has an interactive protocol if 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 holds $\Pr_{r \in \{0,1\}^*} \{(P, V)(x, r) = 1\} \leq \frac{1}{4}$.

Above, the length of r can be polynomially bounded in the length of x.

(b) The class $\mathrm{IP}_\mathbb{R}$ is the class of all decision problems $L \subseteq \mathbb{R}^\infty$ which have an interactive protocol.

The real number complexity class most important for our considerations here was introduced and studied by Cucker and Briquel in [8] and is denoted by $\mathrm{MA}\exists_\mathbb{R}$. Starting point of defining it is the fact that over the reals classes which are classically defined or characterized using space resources turn out to have a more subtle relation among each other than they do classically. Taken alone, space resources have no meaning at all; each decision problem can be decided in linear space using an elementary coding trick [11]. As consequence, for many equivalent characterizations especially of the class PSPACE in classical complexity it is unclear what they should become in the real number framework. Recall that PAR, PSPACE, PAT, and IP, denoting the classes of languages acceptable by parallel polynomial time with exponentially many processors, in polynomial space, in polynomial alternating time, and by interactive proofs, respectively, all are the same in Turing complexity (see the textbook [1] for references and proofs). In contrast, over $\mathbb{R}$ it is known that the real counterparts of the first three classes mentioned above satisfy $\mathrm{PAR}_\mathbb{R} \subsetneq \mathrm{PSPACE}_\mathbb{R} \subseteq \mathrm{PAT}_\mathbb{R}$, where $\mathrm{PSPACE}_\mathbb{R}$ denotes the class of real decision problems decidable by an algorithm using both exponential time and polynomial space[1] and the other two classes are defined by extending the classical definitions straightforwardly, see [4,7]. As a consequence, if a new class like $\mathrm{IP}_\mathbb{R}$ is studied which classically gives yet another characterization of PSPACE via Shamir's famous result [12], it is not at all obvious where

[1] The simultaneous requirement of exponential time and polynomial space excludes the above mentioned coding trick from [11] and makes the definition meaningful.

it has to be located over the reals. The above chain of inclusions also gives the option to define new classes which do not make much sense over finite alphabets. This is precisely where [8] starts by defining classes which can be located between $\mathrm{PSPACE}_{\mathbb{R}}$ and $\mathrm{PAT}_{\mathbb{R}}$. The class $\mathrm{MA}\exists_{\mathbb{R}}$ is one such and turns out to be important for interactive protocols. It at least gives a non-trivial upper bound for $\mathrm{IP}_{\mathbb{R}}$, just indicating the latter to be likely weaker than $\mathrm{PAT}_{\mathbb{R}}$.

Definition 2 ([8]).

(a) The class $\mathrm{MA}\exists_{\mathbb{R}}$ *consists of all decision problems* $L \subseteq \mathbb{R}^\infty$ *for which there exists a problem* $B \in \mathrm{P}_{\mathbb{R}}$ *together with a polynomial* p *such that an* $x \in \mathbb{R}^\infty$ *belongs to* L *if and only if the following formula holds:* $\forall_B z_1 \exists_{\mathbb{R}} y_1 \ldots \forall_B z_{p(|x|)} \exists_{\mathbb{R}} y_{p(|x|)} (x, y, z) \in B$. *The subscripts* $B, \mathbb{R}$ *for quantifiers indicate whether a quantified variable ranges over* $B := \{0, 1\}$ *or* $\mathbb{R}$, *respectively.*

(b) The class $\mathrm{MA}\forall_{\mathbb{R}}$ *consists of the complements of problems in* $\mathrm{MA}\exists_{\mathbb{R}}$. *Thus, a language* $L \in \mathrm{MA}\forall_{\mathbb{R}}$ *contains precisely the points* x *for which a formula of the following form holds:* $\exists_B z_1 \forall_{\mathbb{R}} y_1 \ldots \exists_B z_{p(|x|)} \forall_{\mathbb{R}} y_{p(|x|)} (x, y, z) \in B$.

The following easy lemma is used later on.

Lemma 1. *Let* $\tilde{B} \in \mathrm{MA}\exists_{\mathbb{R}}$ *and* p *be a polynomial. Then the set* $L \subseteq \mathbb{R}^\infty$ *of all* x *such that* $\forall_B y_1 \exists_{\mathbb{R}} z_1 \ldots \forall_B y_{p(|x|)} \exists_{\mathbb{R}} z_{p(|x|)} (x, y, z) \in \tilde{B}$ *belongs to* $\mathrm{MA}\exists_{\mathbb{R}}$, *i.e., if in the definition of* $\mathrm{MA}\exists_{\mathbb{R}}$ *condition* $B \in \mathrm{P}_{\mathbb{R}}$ *is replaced by* $B \in \mathrm{MA}\exists_{\mathbb{R}}$ *we stay within* $\mathrm{MA}\exists_{\mathbb{R}}$[2].

Proof. The quantifier structure related to the definition of $\mathrm{MA}\exists_{\mathbb{R}}$ guarantees the existence of a polynomial q and a problem $B \in \mathrm{P}_{\mathbb{R}}$ such that $(x, y, z) \in \tilde{B} \iff \forall_B s_1 \exists_{\mathbb{R}} t_1 \forall_B s_2 \ldots \forall_B s_{q(|(x,y,z)|)} \exists_{\mathbb{R}} t_{q(|(x,y,z)|)} (x, y, z, s, t) \in B$. So $x \in L$ iff $\forall_B y_1 \exists_{\mathbb{R}} z_1 \ldots \forall_B y_{p(|x|)} \exists_{\mathbb{R}} z_{p(|x|)} \forall_B s_1 \exists_{\mathbb{R}} t_1 \forall_B s_2 \ldots \forall_B s_{q(|(x,y,z)|)} \exists_{\mathbb{R}} t_{q(|(x,y,z)|)} (x, y, z, s, t) \in B$. Since the lengths of both y and z are polynomially bounded in $|x|$ this holds as well for the lengths of s and t. Thus, L has the required representation. □

One of the main results in [8] is proving the inclusion of $\mathrm{PSPACE}_{\mathbb{R}}$ both in $\mathrm{MA}\exists_{\mathbb{R}}$ and $\mathrm{MA}\forall_{\mathbb{R}}$, which in turn implies the strict inclusion of $\mathrm{PAR}_{\mathbb{R}}$ in both latter classes.

Theorem 1 ([8]). $\mathrm{PAR}_{\mathbb{R}} \subsetneq \mathrm{PSPACE}_{\mathbb{R}} \subseteq \mathrm{MA}\exists_{\mathbb{R}} \cap \mathrm{MA}\forall_{\mathbb{R}}$ *and* $\mathrm{MA}\exists_{\mathbb{R}} \cup \mathrm{MA}\forall_{\mathbb{R}} \subseteq \mathrm{PAT}_{\mathbb{R}}$.

Above, the second containment is trivial because by definition of $\mathrm{PAT}_{\mathbb{R}}$ the alternating quantifiers in a defining formula all range over $\mathbb{R}$. The relation between $\mathrm{MA}\exists_{\mathbb{R}}$ and $\mathrm{MA}\forall_{\mathbb{R}}$ currently is unknown as is the question whether one of the two is strictly contained in $\mathrm{PAT}_{\mathbb{R}}$. In the next section we show our main result: $\mathrm{MA}\exists_{\mathbb{R}}$ is an upper bound for $\mathrm{IP}_{\mathbb{R}}$.

[2] This of course only makes sense after $\mathrm{MA}\exists_{\mathbb{R}}$ has been defined precisely.

3 Upper and Lower Bounds for $\mathrm{IP}_{\mathbb{R}}$

In this section we prove the main result of this paper, an upper bound for the class $\mathrm{IP}_{\mathbb{R}}$. In addition, we deal with a few examples for which an interactive proof can be designed. Unfortunately, our results are far from a characterization of $\mathrm{IP}_{\mathbb{R}}$ like Shamir's one in the Turing model. Nevertheless, even the bounds presented here seem not at all obvious and require some efforts. As we shall see, a main obstacle for getting better lower bounds is the fact that now we deal with an uncountable space of information underlying the communications. We shall comment on the results at the end.

3.1 Upper Bound: Recursive Evaluation of Verifier Action

We want to show the inclusion $\mathrm{IP}_{\mathbb{R}} \subseteq \mathrm{MA}\exists_{\mathbb{R}}$. The proof is based on combining a result in [9] with parts of the proof of Theorem 1 adjusted accordingly. The former gives a recursive procedure for computing the number of random vectors under which a verifier accepts an interactive protocol; in the setting of [9] the procedure runs in additive parallel polynomial time. In our setting, however, due to the fact that in particular the prover can send reals the corresponding procedure is not bounded by $\mathrm{PAR}_{\mathbb{R}}$, as has been shown in [9], see also comments below. Instead, dealing with the real information sent introduces real quantifier elimination as part of the task to compute the number of successful random vectors. This naturally leads to considering the class $\mathrm{MA}\exists_{\mathbb{R}}$.

To start with let a verifier V and a prover P be given such that a language $L \subseteq \mathbb{R}^\infty$ has an interactive protocol established by (P, V). In order to decide for an input x whether $x \in L$ it is sufficient to count the number of random strings that cause V to accept x and check whether this number is larger than $\frac{1}{2}$ of the strings. We want to show that this task can be accomplished within class $\mathrm{MA}\exists_{\mathbb{R}}$. Towards this aim below we recall a recursive procedure in [9] for doing this counting and adapt it to our framework. Note that very similar results were already used in the classical discrete setting.

For technical reasons in the proofs below we first need the notion of a normalized real protocol.

Definition 3. *(a) A real protocol (P, V) is called normalized if the following conditions are satisfied:*

- *for input $x \in \mathbb{R}^n, n \in \mathbb{N}$ there are precisely $p(n)$ rounds of communication for a fixed polynomial p only depending on V;*
- *between receiving a real from P and sending a real back to P the verifier V generates precisely one random bit.*

(b) If in a normalized protocol the verifier generates $'0'$ as random bit in round i and then computes a real w_i as the one to be sent to the prover next, we denote it by $w_i(0)$; similarly for $w_i(1)$ if a $'1'$ was generated.

Notational Convention: For a normalized verifier round i starts with generating random bit r_i, then V computes deterministically $w_i \in \mathbb{R}$ and sends it to P in order to receive a $p_i \in \mathbb{R}$. We shall reflect this order of information flow also below in the arguments of some important functions.

It is easy to see that without loss of generality we can assume each protocol to be normalized.

Definition 4 (cf. [9]). *Let (P,V) be a normalized real protocol. For an input $x \in \mathbb{R}^\infty$ suppose the interaction to last $m = poly(|x|)$ rounds. Let $r = (r_1,\ldots,r_m) \in \{0,1\}^*$ be a sequence of random bits V generates during its computation and let $1 \leq j \leq m$.*

(a) For $w_1,\ldots,w_j,p_1,\ldots,p_j \in \mathbb{R}$ denote by $Pass(x,r_1,w_1,p_1,r_2,\ldots,r_j,w_j)$ the relation expressing that for j rounds, given x as input and r_i as random bit in round i the w_i's are the correct data sent by V if P answers with p_i for $1 \leq i \leq j$. Thus, $Pass(x,r_1,w_1,p_1,r_2,\ldots,r_j,w_j) \Leftrightarrow V(x,r_1,w_1,p_1,\ldots,w_{i-1},p_{i-1},r_i) = w_i$ for all $1 \leq i \leq j$.

(b) Define functions Q_j, W_j as follows: $Q_j(x,w_1,p_1,w_2,\ldots,p_{j-1}) := \max |\{r \in \{0,1\}^m \mid Pass(x,r_1,w_1,p_1,\ldots,r_{j-1},w_{j-1}) \wedge w_m = 1\}|$ and $W_j(x,w_1,p_1,w_2,\ldots,p_{j-1},w_j) := \max|\{r \in \{0,1\}^m \mid Pass(x,r_1,w_1,p_1,\ldots,p_{j-1},r_j,w_j) \wedge w_m = 1\}|$. In both cases the max *is taken over all provers that give responses $p_1,\ldots,p_{j-1} \in \mathbb{R}$ as answers to questions $w_1,\ldots,w_{j-1}$ sent by V on input x.*

The only difference between this definition and the corresponding one in [9] is that the w_i and p_i are reals. However, this makes the algorithm behind the next statement more difficult because additional existential quantifiers taking care of the role of the p_i's are needed. Because we deal with normalized protocols the influence of the w_i's now being reals can be controlled without additional quantifiers ranging over $\mathbb{R}$. They basically are determined by the actually generated (discrete) random bit and the previous information of the protocol. This finally is the reason why $\mathrm{MA}\exists_{\mathbb{R}}$ plays an important role.

Lemma 2 (cf. [9], Adapted to Normalized Protocols). *Let $x \in \mathbb{R}^\infty$ and (P,V) be a normalized real protocol accepting a language L. Let m be the number of random bits generated by V during computation on x. Then for all $1 \leq j \leq m-1$*

(a) x is accepted iff $Q_1(x) > 2^{m-1}$;
(b) $Q_j(x,w_1,\ldots,p_{j-1}) = W_j(x,w_1,\ldots,p_{j-1},w_j(0)) + W_j(x,w_1,\ldots,p_{j-1},w_j(1))$;
(c) $W_j(x,w_1,\ldots,p_{j-1},w_j) = \max\limits_{p_j \in \mathbb{R}} Q_{j+1}(x,w_1,\ldots,p_{j-1},w_j,p_j)$;
(d) $W_m(x,w_1,\ldots,p_{m-1},w_m) = |\{r \in \{0,1\}^m \mid Pass(x,r_1,w_1,p_1,r_2,\ldots,p_{m-1},r_m,1)\}|$.

Proof. Except for small changes the proof is similar to the corresponding one in [9]. Item (a) follows from the definition of Q_1 and that of (P,V) accepting an input x. For (b) first note that the verifier at most generates two different

real values for w_j given x, r, and the p_i, w_i for $1 \leq i \leq j-1$, namely $w_j(0)$ and $w_j(1)$. In this sense the choice of possible w_j's still is discrete. For each prover P used as candidate for the max in the definition of Q_j it is easy to see that the number of accepting r it yields is at most as large as the sum on the right hand side. This holds because the same P is a candidate for both max computations related to $W_j(x, \ldots, w_j(0))$ and $W_j(x, \ldots, w_j(1))$. Vice versa, if P_0, P_1 denote the optimal provers in the definitions of $W_j(x, \ldots, w_j(0))$ and $W_j(x, \ldots, w_j(1))$, respectively, an optimal prover for the max in Q_j behaves like P_0 if $r_j = 0$ and $w_j(0)$ was sent by the verifier and like P_1 if $r_j = 1$ and $w_j(1)$ was sent. This proves the equality.

For (c) the max defining W_j asks for the best continuation of the protocol when $x, r_1, \ldots, r_{j-1}, p_1, \ldots, p_{j-1}, w_1, \ldots, w_j$ are fixed. The next portion of data that can be chosen to achieve the maximum is $p_j \in \mathbb{R}$. The right hand side just asks for the optimal one. Thus both sides are equal. Item (d) holds because the part $(p_1, \ldots, p_{m-1})$ of the argument W_m already fixes the prover. □

The lemma is used in order to compute recursively whether an input x is accepted, i.e., computing $Q_1(x)$ and deciding whether it is larger than 2^{m-1}. In the discrete setting this can be achieved in parallel polynomial time. In our situation, however, due to Theorem 2 in [9] this is not possible;[3] there are problems provably in $\mathrm{IP}_{\mathbb{R}}$ but not in $\mathrm{PAR}_{\mathbb{R}}$. Therefore, the involved max computations increase the problem's difficulty because maximization over an uncountable domain is required. The formal description of the resulting problem introduces $\exists_{\mathbb{R}}$-quantifiers and thus leads to $\mathrm{MA}\exists_{\mathbb{R}}$ as an upper bound for $\mathrm{IP}_{\mathbb{R}}$.

Theorem 2. $\mathrm{IP}_{\mathbb{R}} \subseteq \mathrm{MA}\exists_{\mathbb{R}}$.

Proof. Let $L \in \mathrm{IP}_{\mathbb{R}}$ and (P, V) a corresponding normalized protocol for L. For an input $x \in \mathbb{R}^n$ let $m = poly(n)$ denote the polynomial number of rounds and generated random bits of the interaction. In order to use the above lemma algorithmically it is necessary to compute the maxima occuring in the statements. Though the prover's answers range over real numbers the maxima are integers. Consider $Q_m(x, w_1, p_1, w_2, \ldots, p_{m-1}) = |\{r \in \{0,1\}^m \mid Pass(x, r_1, w_1, p_1, r_2, \ w_2, \ldots, p_{m-1})\}|$. Obviously, both Q_m and the predicate "$Q_m(x, w_1, \ldots, p_{m-1}) = s$ for given $s \in \{0, 1, \ldots, 2^m\}$" are computable in $\mathrm{PAR}_{\mathbb{R}}$ because for every single r the simulation of (P, V) on x using r needs polynomial time. Now for each $s \in \{0, 1, \ldots, 2^m\}$ the predicate $\exists p_{m-1} \in \mathbb{R} : \ Q_m(x, w_1, \ldots, p_{m-2}, w_{m-1}, p_{m-1}) = s$ belongs to $\mathrm{MA}\exists_{\mathbb{R}}$: First, Theorem 1 implies that the inner predicate "$Q_m = s$" is in $\mathrm{MA}\exists_{\mathbb{R}}$, then Lemma 1 shows that this class is not left. In order to compute W_{m-1} we can in parallel compute for each $s \in \{0, 1, \ldots, 2^m\}$ whether '$\exists p_{m-1} \in \mathbb{R} : Q_m = s$' holds and finally extract the maximal s for which this is true in polynomial time. Again by Lemma 1 it follows that the predicate $W_{m-1} = s$ can be decided in $\mathrm{MA}\exists_{\mathbb{R}}$. The same holds for the predicate $Q_{m-1}(q, w_1, \ldots, p_{m-1}) = s$ since according to

[3] Though formally the classes in [9] are defined a bit differently it is easy to see that their protocols used to prove the theorem fit into $\mathrm{IP}_{\mathbb{R}}$.

part (b) of Lemma 2 it can be computed using a sum of W_{m-1} when the last component once is $w_{m-1}(0)$ and once $w_{m-1}(1)$.

We continue along the recursion behind Lemma 2. Since its depth m is polynomially bounded in $n = |x|$, by precisely the same arguments as above we see that all predicates $Q_1 = s$ (or, similarly, $Q_1 > s$) for $s \in \{0, 1, \ldots, 2^m\}$ belong to $\mathrm{MA}\exists_{\mathbb{R}}$; the structure $\forall_B \exists_{\mathbb{R}}$ of quantifier prefixes remains the same and for each of the m rounds only a polynomial number of quantifiers is added. Finally, $x \in L$ iff $Q_1 > 2^{m-1}$ finishes the proof. □

3.2 Lower Bounds

In this subsection we report on ongoing research to obtain meaningful lower bounds for $\mathrm{IP}_{\mathbb{R}}$. However, it currently is more a discussion of problems and interesting open questions than a completed project. In particular, so far we have not been able to give a characterization of $\mathrm{IP}_{\mathbb{R}}$ analogue to Shamir's result. Below, we discuss where new difficulties arise and what might be promising ways to go.

Let us first give some interactive proofs for certain restricted subclasses of problems. These results might already shed some light on the difficulties faced when trying to generalize the classical methods to design IP's to the real model.

A major problem here seems to be to obtain suitable arithmetizations of the problems considered in order to apply similar techniques. $\mathrm{PAT}_{\mathbb{R}}, \mathrm{MA}\exists_{\mathbb{R}}, \mathrm{MA}\forall_{\mathbb{R}}$ are classes defined by quantifying a problem $B \in \mathrm{P}_{\mathbb{R}}$ using different sequences of quantifiers of different structures. Another example of such a class is $\mathrm{DPAT}_{\mathbb{R}}$, where all quantifiers are Boolean. It seems natural to expect that at least for sequences of Boolean quantifiers the classical techniques could be adopted. However, this is not obviously true, the reason being the need of a suitable arithmetization of properties $B \in \mathrm{P}_{\mathbb{R}}$. In the following we consider certain subclasses obtained by restricting parts of the general problem definition: either the quantifier structure or the condition in $\mathrm{P}_{\mathbb{R}}$ or both. We shall investigate some cases for which interactive protocols can be designed.

Definition 5. *(a) We denote by* $\mathrm{MA}\forall_{\mathbb{R}}^{=0}$ *the subclass of problems in* $\mathrm{MA}\forall_{\mathbb{R}}$ *where* $B \in \mathrm{P}_{\mathbb{R}}$ *can be chosen to be of the following particular form: There is a multivariate polynomial* F_x *such that* $(x, y, z) \in B$ *if and only if* $F_x(y, z) = 0$. *Moreover, given* (x, y, z) *the value* $F_x(y, z)$ *can be computed in polynomial time in the size of* x.

(b) A problem S *is in class* $\mathrm{DPAT}_{\mathbb{R}}$ *if there is a polynomial* p *and another problem* $B \in \mathrm{P}_{\mathbb{R}}$ *such that* $x \in S$ *if and only if* $\forall_B z_1 \exists_B z_2 \ldots Q_{p(|x|)} z_{p(|x|)} (x, z_1, ..., z_{p(|x|)}) \in B$, *where* $Q_{p(|x|)} \in \{\exists_B, \forall_B\}$.

(c) A problem $S \in \mathrm{DPAT}_{\mathbb{R}}$ *belongs to class* $\mathrm{DPAT}_{\mathbb{R}}^{=0}$ *if the condition* $(x, z) \in B$ *in part (b) has the particular form* $F_x(z) = 0$ *for a polynomial* F_x *that can be evaluated in polynomial time in* $|x|$. *Similarly, class* $\mathrm{DPAT}_{\mathbb{R}}^{\neq 0}$ *consists of problems where this condition reads* $F_x(z) \neq 0$.

For the class $\mathrm{MA}\forall_{\mathbb{R}}^{=0}$ of problems defined above we obtain interactive protocols basically by applying the classical Schwartz-Zippel lemma.

Proposition 1. *It holds* $\mathrm{MA}\forall_{\mathbb{R}}^{=0} \subseteq \mathrm{IP}_{\mathbb{R}}$.

Problems in class $\mathrm{DPAT}_{\mathbb{R}}$ are defined using Boolean quantifier prefixes only. Thus, one might expect that the classical discrete technique for designing interactive proofs suffices. However, the problem seems to be finding a suitable arithmetization of the formula. For the subclasses defined above this is possible.

Proposition 2. *It holds* $\mathrm{DPAT}_{\mathbb{R}}^{=0} \subset \mathrm{IP}_{\mathbb{R}}$ *and* $\mathrm{DPAT}_{\mathbb{R}}^{\neq 0} \subset \mathrm{IP}_{\mathbb{R}}$.

Another possible way to extend the class of problems that have a real interactive protocol is the examination of oracle computations and counting problems. In [10] an interactive protocol for verifying the value of a permanent of a 0-1-matrix was given (before Shamir's result was known). Together with Toda's theorem that the polynomial hierarchy PH is included in $P^{\#P}$ and the $\#P$-completeness of the permanent computation this implies an interactive protocol for all problems in the polynomial hierarchy. The protocol for the permanent, as for example described in [1], works as well for real matrices in the BSS model. This implies that real problems that can be decided by a polynomial time BSS algorithm having access to an oracle computing the permanent of real number matrices, i.e., all problems in class $\mathrm{P}_{\mathbb{R}}^{Perm}$, belong to $\mathrm{IP}_{\mathbb{R}}$. However, it is not known whether the permanent plays a similar role for real counting problems as it does in the Turing model. This is an active field of research. Basu and Zell [3] have given a real analogue of Toda's theorem. Instead of the permanent in this approach the computation of so-called Betti numbers of semi-algebraic sets plays a crucial role. The latter express certain topological properties of semi-algebraic sets. They seem to be even more difficult to handle than permanent computations. An intensive study of counting problems has been performed in [5,6] for both the additive and the full real number model. Further topological quantities that turn out to be important are, for example, the topological degree and the Euler characteristic of a set. In both papers several characterizations of real complexity classes via oracle computations as well as completeness results are given. The results in the additive setting actually can be used to prove again the main result of [9].

Theorem 3 ([9]). *In the additive real BSS model the class* $PAR_{\mathbb{R},+}$ *of problems decidable in parallel polynomial time equals the class* $BIP_{\mathbb{R},+}$ *of problems that admit an additive interactive protocol only exchanging boolean messages.*

The prove is technically very similar to the one in [9] in that a crucial inclusion $\mathrm{PAR}_{\mathbb{R},+} \subseteq \mathrm{P}_{\mathbb{R}_+}^{\mathrm{PSPACE}}$ is shown using the existence of small rational points in certain point location tasks. A similar result is central in [9]. This gives the possibility to involve discrete oracles which then can be handled using the classical protocol by Shamir. It is not known whether in the full model discrete oracles play a similarly important role. But the above reasoning makes it interesting to study which real (counting) functions bearing a high complexity can be computed by an interactive protocol in order to use it as an oracle. Another example are so-called resultant functions which are polynomials built from the coefficients

of certain polynomial systems and crucial in some of the currently best known algorithms for dealing with the existential theory over the reals like determinants are for the solution of linear systems. Thus we have

Problem 1: Can any of the following problems be solved by an interactive protocol in the full BSS model: given a semi-algebraic set $S \subseteq \mathbb{R}^n$ via a system of polynomial (in-)equalities and a number $k \in \mathbb{N}_0$, verify that the sum of the Betti-numbers of S or the degree or the Euler-characteristic of S equals k. What about verifying the value of resultant polynomials by an interactive protocol?

Even if it is unclear whether positive answers would give the intended characterization of $\mathrm{IP}_{\mathbb{R}}$ it would be a significant step forward. For example, existence of such protocols for the Euler characteristic or the Betti numbers would imply that co-$\mathrm{NP}_{\mathbb{R}} \subseteq \mathrm{IP}_{\mathbb{R}}$ because the latter can be solved using a polynomial time oracle computation that has access to evaluating those function.

Unfortunately, at the moment we do not know how to design an interactive protocol for co-$\mathrm{NP}_{\mathbb{R}}$. It seems unlikely that $\mathrm{MA}\exists_{\mathbb{R}} = \mathrm{MA}\forall_{\mathbb{R}}$. Thus, if $\mathrm{MA}\exists_{\mathbb{R}}$ turns out to equal $\mathrm{IP}_{\mathbb{R}}$ it would not be obvious whether $\mathrm{IP}_{\mathbb{R}}$ is closed under complementation. Classically, this of course holds.

Problem 2: Is $\mathrm{IP}_{\mathbb{R}}$ closed under complementation?

References

1. Arora, S., Barak, B.: Computational Complexity: A Modern Approach. Cambridge University Press, Cambridge (2009)
2. Baartse, M., Meer, K.: The PCP theorem for NP over the reals. Found. Comput. Math. Springer. doi:10.1007/s10208-014-9188-x
3. Basu, S., Zell, T.: Polynomial hierarchy, Betti numbers, and a real analogue of Toda's theorem. Found. Comput. Math. **10**(4), 429–454 (2010)
4. Blum, L., Cucker, F., Shub, M., Smale, S.: Complexity and Real Computation. Springer, New York (1998)
5. Bürgisser, P., Cucker, F.: Counting complexity classes for numeric computations I: semilinear sets. SIAM J. Comput. **33**(1), 227–260 (2003)
6. Bürgisser, P., Cucker, F.: Counting complexity classes for numeric computations. II. algebraic and semialgebraic sets. J. Complex. **22**(2), 147–191 (2006)
7. Cucker, F.: On the complexity of quantifier elimination: the structural approach. Comput. J. **36**(5), 400–408 (1993)
8. Cucker, F., Briquel, I.: A note on parallel and alternating time. J. Complex. **23**, 594–602 (2007)
9. Ivanov, S., de Rougemont, M.: Interactive protocols on the reals. Comput. Complex. **8**, 330–345 (1999)
10. Lund, C., Fortnow, L., Karloff, H., Nisan, N.: Algebraic methods for interactive proof systems. J. ACM **39**(4), 859–868 (1992)
11. Michaux, C.: Une remarque à propos des machines sur $\mathbb{R}$ introduites par Blum. Shub et Smale. C.R. Acad. Sci. Paris, t. 309, Série I, pp. 435–437 (1989)
12. Shamir, A.: IP = PSPACE. J. ACM **39**(4), 869–877 (1992)