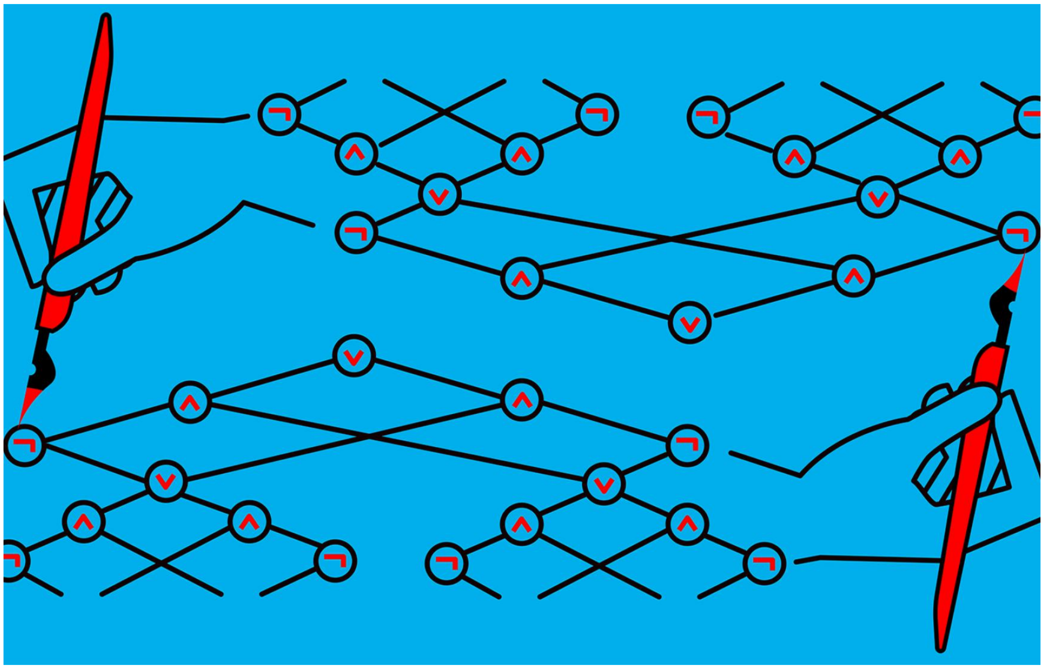
Meta-Complexity



Open Problems

Simons Institute for the Theory of Computing

Spring 2023

Contents

NP-Hardness of Meta-Complexity of Average-Case Complexity

Submitted by Hanlin Ren on 28/January/2023.

Description. Many meta-complexity problems are known to be NP-hard (see, e.g., [Ila20a; Ila20b; ILO20]). In these NP-hardness reductions, the Yes instances generated are computable by a small circuit, and the No instances are *worst-case* hard against the same class of circuits. Can we prove the NP-hardness of meta-complexity problems where the No instances are *average-case* hard against the corresponding circuit family? For example, while the NP-hardness of DNF-MCSP is already known [All+06], the following problem remains open:

Problem 1. *Is the following problem* NP*-hard under quasi-polynomial time randomized reductions?*

- *Input: the truth table of a function* $f : \{0,1\}^n \to \{0,1\}$ *and a size parameter* s.
- *Yes instances: there is a size-*s *DNF that computes* f *on the worst case.*
- *No instances: for every size-*s *DNF* F, $\Pr_{x \leftarrow \{0,1\}^n}[F(x) = f(x)] \le 0.9$.

The second question concerns *polylog*-time-bounded Kolmogorov complexity. Despite being (seemingly) less natural, it has an interesting motivation, namely to capture the PCP theorem by meta-complexity.

In what follows, let U be a universal random access machine. Given a machine d, some auxiliary input x, and a time bound t, $U^{d,x}(t) = 1$ if and only if the machine d accepts the input x in time t. We assume that d has random access to x, and each random access requires exactly $\lceil \log |x| \rceil$ time (to write down the address i if we want to access x_i). We also assume that U has random access to both d and x (so if it is the case that $|d| > t$ but d still terminates in t steps, U can also simulate everything in $\mathrm{poly}(t)$ steps without reading d entirely).

Given this model, we can define the *conditional polylog-time-bounded Kolmogorov complexity* as follows: Let $t = \mathrm{polylog}(n)$, $x \in \{0,1\}^n$, $y \in \{0,1\}^{\mathrm{poly}(n)}$, then $\mathrm{K}^t(x \mid y)$ is the minimum length $(|d| + |z|)$ over all descriptions (d, z) such that for every $i \in [n+1]$, $U^{(y,z,i)}(d,t) = x_i$. (Assume $x_{n+1} = \star$.)

Problem 2. *Let* $t = \mathrm{polylog}(n)$*. Is the following problem* NP*-hard under polynomial time randomized reductions?*

- *Input: strings* $x \in \{0,1\}^n$, $y \in \{0,1\}^{\mathrm{poly}(n)}$ *and a size parameter* s.
- *Yes instances:* $\mathrm{K}^t(x \mid y) \le s$.
- *No instances: for every* x' *that agrees with* x *on* 90% *fraction of indices,* $\mathrm{K}^t(x' \mid y) > s$.

Denote Ave-cK$^{\mathrm{polylog}}$ as the meta-complexity problem in Problem 2. We observe that Ave-cK$^{\mathrm{polylog}}$ has a simple PCP system: the prover sends a length-s description (d, z) to the verifier, the verifier randomly picks some index $i \leftarrow [n]$, runs $U^{(y,z,i)}(d, \mathrm{polylog}(n))$, and sees if it equals to x_i. Therefore, an NP-hardness proof of Ave-cK$^{\mathrm{polylog}}$ would be a "meta-complexity analog" of the PCP theorem.

Additional comments: Hirahara [Hir22] provided an interesting example of "meta-complexity of average-case complexity". In particular, Hirahara showed that given the description of a sampler $\mathcal{D}$ that produces a pair $(x, y) \in \{0,1\}^n \times \{0,1\}$, it is NP-hard to distinguish between the case that (1) there is a circuit C of size s such that $\Pr_{(x,y)\leftarrow\mathcal{D}}[C(x) = y] = 1$ and that (2) for any circuit C of size $s \cdot n^{1/\log^{O(1)}\log n}$, $\Pr_{(x,y)\leftarrow\mathcal{D}}[C(x) = y] \le 0.51$.

References

[All+06] Eric Allender, Lisa Hellerstein, Paul McCabe, Toniann Pitassi, and Michael E. Saks. “Minimizing DNF Formulas and AC^0_d Circuits Given a Truth Table”. In: *Computational Complexity Conference*. IEEE Computer Society, 2006, pp. 237–251.

[Hir22] Shuichi Hirahara. “NP-Hardness of Learning Programs and Partial MCSP”. In: *FOCS*. IEEE, 2022, pp. 968–979.

[Ila20a] Rahul Ilango. “Approaching MCSP from Above and Below: Hardness for a Conditional Variant and $\mathsf{AC}^0[p]$”. In: *ITCS*. Vol. 151. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 34:1–34:26.

[Ila20b] Rahul Ilango. “Constant Depth Formula and Partial Function Versions of MCSP are Hard”. In: *FOCS*. IEEE, 2020, pp. 424–433.

[ILO20] Rahul Ilango, Bruno Loff, and Igor Carboni Oliveira. “NP-Hardness of Circuit Minimization for Multi-Output Functions”. In: *Computational Complexity Conference*. Vol. 169. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 22:1–22:36.

Witnessing of $\mathsf{NP} \not\subseteq \mathsf{P/poly}$

Submitted by Jan Pich and Rahul Santhanam on 30 January 2023.

Problem 3. *Fix a constant $k \geq 1$. Suppose that for each sufficiently big n, no circuit with n inputs and size n^k finds a satisfying assignment for each satisfiable propositional formula of size n, i.e. no n^k-size circuit solves the search version of SAT. Can we witness this assumption 'feasibly' by a p-time function f such that for each sufficiently big n, for each n^k-size circuit C with n inputs and $\leq n$ outputs, $f(C)$ outputs a formula ϕ of size n together with a satisfying assignment a of ϕ such that $\neg\phi(C(\phi))$?*

Such a 'witnessing' function f exists under the assumption of the existence of a one-way function and a function in E hard for subexponential-size circuits [4]. Is it, however, possible to construct it without assuming more than the assumption we want to witness?

Related work.

Similar kinds of witnessing have been considered before in the literature, using diagonalization techniques [1-3]. Indeed, Bogdanov, Talwar and Wan [2] call a similar feasible witnessing in the uniform setting a "dreambreaker" (following Adam Smith) and show that such a feasible witnessing can be constructed. However, in Problem 3 it is crucial that the witnessing function finds an error on the input length of the given circuit.

References

[1] Atserias A.; *Distinguishing SAT from polynomial-size circuits, through black-box queries*; Computational Complexity Conference (CCC), 2006.

[2] Bogdanov A., Talwar K., Wan A.; *Hard Instances for Satisfiability and Quasi-one-way Functions*; ICS, 2010.

[3] Gutfreund D., Shaltiel R., Ta-Shma A.; *If NP languages are hard in the worst-case then it is easy to find their hard instances*; Computational Complexity, 16(4), 412– 441, 2007.

[4] Müller M., Pich J.; *Feasibly constructive proofs of succinct weak circuit lower bounds*; Annals of Pure and Applied Logic, 2019.

Narrow the Gap in an Impossibility Result

Submitted by Eric Allender on 31/January/2023.

Description. MKTP is the problem of computing the KT complexity of a string: MKTP $= \{(x, i) : KT(x) < i\}$. It is shown in [AHT22, Theorem 35] that there is no projection f such that $z \in$ MKTP implies $K(f(z)) > \frac{4|f(z)|}{5}$ and $z \notin$ MKTP implies $K(f(z)) < \frac{|f(z)|}{5}$.

Problem 4. *Improve this gap from* $[\frac{1}{5}, \frac{4}{5}]$ *to, say,* $[\frac{1}{3}, \frac{2}{3}]$.

Additional comments. Improving it to $[\frac{1}{2} - 2^{\sqrt{\log n}}, \frac{1}{2} + 2^{\sqrt{\log n}}]$ would show NL $\neq$ NP.

References

[AHT22] Eric Allender, Shuichi Hirahara, and Harsha Tirumala. *Kolmogorov Complexity Characterizes Statistical Zero Knowledge.* Tech. rep. TR22-127. To appear in ITCS 2023. Electronic Colloquium on Computational Complexity (ECCC), 2022.

Tractable(?) Open Questions about Partial MCSP

Submitted by Shuichi Hirahara on 3/February/2023.

Description. This is the set of open questions raised in my talk about [Hir22b] at the meta-complexity reading group.

Problem 5. *Prove that* Formula-MCSP* *is reducible to* Formula-MCSP *in subexponential time.*

1. Ilango [Ila21] presented an exponential-time reduction (whose running time depends on the size parameter) from Formula-MCSP* to Formula-MCSP.
2. There are polynomial-time partial-to-total reductions in the case of MCSP for DNF formulas [All+08] and DNF-XOR formulas [HOS18].

Problem 6. *Prove that* Formula-MCSP* *is* NP*-hard.*[1]

Note that [Hir22b] shows NP-hardness of NC^1-MCSP*. Since Uhlig's theorem does not hold for formulas, you need to eliminate the usage of Uhlig's theorem in the proof of [Hir22b].

Problem 7. *Prove that* MCSP *is* NP*-hard under cryptographic assumptions (or any reasonable assumptions).*

Intuitively, the reason why the reduction of [Hir22b] outputs a partial function is that there is no efficient way of checking that the shares are correctly distributed or not by a secret sharing scheme. Cryptographic tools, such as zero knowledge proof systems, might be helpful.

Problem 8. *Prove that* AC^0*-Circuit-*MCSP* *is* NP*-hard. Here, we measure the size of an* AC^0 *circuit by the number of gates in the circuit.*

In the proof of [Hir22b], it is implicitly used that $2^n \approx \mathrm{K}(f) = \widetilde{\Theta}(\mathrm{CC}(f)) = \widetilde{\Theta}(2^n/n)$ holds with high probability over a uniformly random function $f\colon \{0,1\}^n \to \{0,1\}$. Here, $\mathrm{CC}(f)$ denotes the circuit complexity of f and $\mathrm{K}(f)$ denotes the Kolmogorov complexity of f. On the other hand, a uniformly random function has AC^0 circuits of size $\Theta(2^{n/2})$ [Dan96].

Problem 9. *Prove that* GapMINKT$^{*,\mathrm{SAT}}$ *is complete for* Σ_2^p*. Here,* GapMINKT$^{*,\mathrm{SAT}}$ *is the problem of approximating the time-bounded Kolmogorov complexity of a given string* x *up to an additive error of* $O(\log|x|)$.[2]

[Hir22b] shows that GapMINKT* is NP-hard. Problem 9 asks whether this result can be relativized to the SAT oracle. Note that [Ko91] constructed an oracle A under which GapMINKT* is not NP-hard, but A is not SAT.

Problem 9 has a significant consequence to the worst- vs. average-case complexity of PH: If Problem 9 is resolved, then $\mathsf{DistPH} \not\subseteq \mathsf{AvgP}$ unless PH collapses. The reason is as follows:

$$\mathsf{coNP} \leq_m^p \mathsf{NP}^{\mathsf{NP}} = \Sigma_2^p \leq_m^{\mathsf{BPP}} \mathrm{GapMINKT}^{*,\mathrm{SAT}} \leq_m^{\mathsf{NP}} \mathrm{GapMINKT}^{\mathrm{SAT}},$$

where the last reduction follows from a simple partial-to-total reduction [GR22].[3] The last problem GapMINKT$^{\mathrm{SAT}}$ is known to be in P if $\mathsf{DistPH} \subseteq \mathsf{AvgP}$ [Hir20]. Thus, under the assumption that PH is easy on average, we obtain $\mathsf{coNP} \subseteq \mathsf{AM}$.[4]

[1] I conjecture that this is the easiest open question.

[2] The precise definition of GapMINKT$^{\mathrm{SAT}}$ can be found in [Hir22a].

[3] The reduction non-deterministically guesses how to fill out the stars by 0 or 1.

[4] In fact, one can also get $\mathsf{NP} = \mathsf{coNP}$.

References

[All+08] Eric Allender, Lisa Hellerstein, Paul McCabe, Toniann Pitassi, and Michael E. Saks. "Minimizing Disjunctive Normal Form Formulas and AC^0 Circuits Given a Truth Table". In: *SIAM J. Comput.* 38.1 (2008), pp. 63–84.

[Dan96] Vlado Dancik. "Complexity of Boolean Functions Over Bases with Unbounded Fan-In Gates". In: *Inf. Process. Lett.* 57.1 (1996), pp. 31–34.

[GR22] Ludmila Glinskih and Artur Riazanov. "MCSP is Hard for Read-Once Nondeterministic Branching Programs". In: *LATIN 2022: Theoretical Informatics - 15th Latin American Symposium, Guanajuato, Mexico, November 7-11, 2022, Proceedings.* 2022, pp. 626–640.

[Hir20] Shuichi Hirahara. "Characterizing Average-Case Complexity of PH by Worst-Case Meta-Complexity". In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS).* 2020, pp. 50–60.

[Hir22a] Shuichi Hirahara. "Meta-Computational Average-Case Complexity: A New Paradigm Toward Excluding Heuristica". In: *Bull. EATCS* 136 (2022).

[Hir22b] Shuichi Hirahara. "NP-Hardness of Learning Programs and Partial MCSP". In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS).* 2022.

[HOS18] Shuichi Hirahara, Igor Carboni Oliveira, and Rahul Santhanam. "NP-hardness of Minimum Circuit Size Problem for OR-AND-MOD Circuits". In: *Proceedings of the Computational Complexity Conference (CCC).* 2018, 5:1–5:31.

[Ila21] Rahul Ilango. "The Minimum Formula Size Problem is (ETH) Hard". In: *Proceedings of the Symposium on Foundations of Computer Science (FOCS).* 2021, pp. 427–432.

[Ko91] Ker-I Ko. "On the Complexity of Learning Minimum Time-Bounded Turing Machines". In: *SIAM J. Comput.* 20.5 (1991), pp. 962–986.

Refuting Symmetry of Information for rKt

Submitted by Zhenjian Lu on 21/February/2023.

Description. Symmetry of information for time-bounded Kolmogorov complexity states that For every $x, y \in \{0,1\}^*$, $\mathrm{K}(x,y) \geq \mathrm{K}(x) + \mathrm{K}(x \mid y) - O(\log(|x| + |y|))$. It was known that symmetry of information does not hold for Kt complexity *unconditionally* [Ron04], where Kt is a notion of time-bounded Kolmogorov complexity introduced by Levin. A randomized analogue of Kt, called rKt, was defined by Oliveira [Oli19]. It is not hard to show that if one-way functions exist, then symmetry of information does not hold for rKt, but we don't know how to show this unconditionally.

Problem 10. *Refute symmetry of information for* rKt*. That is, show that for every constant* $c \geq 1$*, there are infinitely many* n *and* $x, y \in \{0,1\}^n$ *such that* $\mathrm{rKt}(x,y) \geq \mathrm{rKt}(x) + \mathrm{rKt}(x \mid y) - c \cdot (\log(|x| + |y|))$

References

[Oli19] Igor C. Oliveira. "Randomness and Intractability in Kolmogorov Complexity". In: *International Colloquium on Automata, Languages, and Programming* (ICALP). 2019, 32:1–32:14.

[Ron04] Detlef Ronneburger. "Kolmogorov Complexity and Derandomization". PhD thesis. Rutgers University, 2004.

Open Questions in Relativized Heuristica

Submitted by Mikito Nanashima on 21/February/2023.

Description. I would like to share two open questions in Heuristica for which no relativization barrier result is currently known (namely, relativizing proofs could still work for solving the problems below).

The first question is about the relationships between learning and average-case complexity. In [HN21], we showed that the errorless average-case easiness of NP implies a polynomial-time PAC learner for polynomial-size circuits that works on only unknown $\mathsf{P/poly}$-samplable example distributions. Here, the main difference to the standard *distribution-free* PAC learner is that the time-complexly of our learner needs to be polynomially larger than the time-complexity of the sampling algorithm for example distributions. In addition, we showed that this additional requirement is inevitable for subexponential-time learning in relativized Heuristica by presenting the oracle separation between the average-case easiness of PH and $2^{o(\frac{n}{\log n})}$-time *distribution-free* PAC learning (for linear-size circuits). However, it is still open whether *non-trivial* distribution-free PAC learning is feasible in relativized Heuristica.

Problem 11. *Does* $\mathsf{DistNP} \subseteq \mathsf{AvgP}$ *imply* distribution-free *weak learning for linear-size circuits in time* $2^n/n^{\omega(1)}$*? Or can we show a relativization barrier?*

Even under the stronger assumption that $\mathsf{DistPH} \subseteq \mathsf{AvgP}$, the problem above is open. Note that it is unclear whether the work [Hir21] yields a non-trivial learner because the instance size of sample sets for learning linear size grows $\Omega(n^2)$ for example size n.

The second question is about the relationships between errorless and error-prone average-case complexity. In [HN22], we showed the oracle separation between errorless and error-prone average-case complexity of NP. The proof heavily relies on a property of DNFs (which corresponds to NP-computation), so extending the separation to the average-case error-prone complexity of PH is currently out of reach. Does this mean we can indeed derive errorless average-case easiness of NP from a stronger assumption on error-prone average-case easiness? Or, it is possible to improve the oracle separation result in [HN22]?

Problem 12. *Does* $\mathsf{DistPH} \subseteq \mathsf{HeurP}$ *imply* $\mathsf{DistNP} \subseteq \mathsf{AvgP}$*? Or can we show a relativization barrier?*

The problem above is open even for a weaker consequence that $\mathsf{DistNP} \subseteq \mathsf{AvgSIZE}[2^{O(\frac{n}{\log n})}]$. Personally, I think showing the non-existence of auxiliary-input OWF [OW93] under $\mathsf{DistPH} \subseteq \mathsf{HeurP}$ can be an interesting intermediate challenge towards an affirmative answer for Problem 12.

References

[Hir21] Shuichi Hirahara. "Average-case hardness of NP from exponential worst-case hardness assumptions". In: *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*. Ed. by Samir Khuller and Virginia Vassilevska Williams. ACM, 2021, pp. 292–302.

[HN21] Shuichi Hirahara and Mikito Nanashima. “On Worst-Case Learning in Relativized Heuristica”. In: *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*. IEEE, 2021, pp. 751–758.

[HN22] Shuichi Hirahara and Mikito Nanashima. “Finding Errorless Pessiland in Error-Prone Heuristica”. In: *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*. Ed. by Shachar Lovett. Vol. 234. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 25:1–25:28.

[OW93] Rafail Ostrovsky and Avi Wigderson. “One-Way Fuctions are Essential for Non-Trivial Zero-Knowledge”. In: *Second Israel Symposium on Theory of Computing Systems, ISTCS 1993, Natanya, Israel, June 7-9, 1993, Proceedings*. IEEE Computer Society, 1993, pp. 3–17.

Characterizing Exponentially hard OWFs

Submitted by Yanyi Liu on 21/Feb/2023.

Description. Exponentially hard one-way functions (OWFs) are polynomial time computable functions that are hard to invert by exponential time attackers.

Problem 13. *Find a natural computational assumption that characterizes the existence of exponentially hard OWFs.*

Additional comments Any natural hardness assumptions (whether or not related to meta-complexity) are welcome.

Related work. It is shown in [LP21] that average-case hardness of time-bounded Kolmogorov complexity problems characterizes OWFs with hardness ranging from polynomial to subexponential. Extending this result to the exponential hardness regime requires proving a strong version of Yao's hardness amplification lemma.

References

[LP21] Yanyi Liu and Rafael Pass. "Cryptography from Sublinear Time Hardness of Time-bounded Kolmogorov Complexity". In: *STOC*. 2021.

Degree-2-Avoid

Submitted by Karthik Gajulapalli and Sidhant Saraogi on 23/March/2022.

Description. Degree-2-Avoid is a version of the range avoidance problem, where given a circuit with n inputs, and $m > n$ outputs, each output bit is computed by a degree-2 polynomial over its input bits.

Problem 14. *Are there algorithmic techniques to solve degree-*2-Avoid *that do not use a reduction to* NC_0^3-Avoid*?*

Related work. In [Gaj+23] it was shown that degree-2-Avoid can encode hard combinatorial objects like rigid-matrices. While randomized encodings [AIK06] provide a way of reducing degree-2-Avoid to NC_0^3-Avoid, such reductions end up killing the stretch of the resulting instance thus making it a much harder problem to solve. In fact, solving degree-2-Avoid with super-linear stretch $m = n^{4/3}$ would provide construction of rigid matrices that beat the best known constructions by [AC19] [Bha+20] and solving degree-2-Avoid for stretch $m = 2n$ would imply a super-linear circuit lowerbound due to Valiant's program [Val77].

References

[AC19] Josh Alman and Lijie Chen. "Efficient construction of rigid matrices using an NP oracle". In: *FOCS*. 2019.

[AIK06] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. "Cryptography in NC^0". In: *SIAM Journal on Computing* 36.4 (2006), pp. 845–888.

[Bha+20] Amey Bhangale, Prahladh Harsha, Orr Paradise, and Avishay Tal. "Rigid Matrices From Rectangular PCPs". In: *FOCS*. 2020.

[Gaj+23] Karthik Gajulapalli, Alexander Golovnev, Satyajeet Nagargoje, and Sidhant Saraogi. "Range Avoidance for Constant-Depth Circuits: Hardness and Algorithms". In: *arXiv preprint arXiv:2303.05044* (2023).

[Val77] Leslie G. Valiant. "Graph-theoretic arguments in low-level complexity". In: *MFCS*. 1977.

Which theories prove $\mathsf{PIT} \in \mathsf{P/poly}$?

Submitted by Albert Atserias on 28/March/2023.

Description.

The Algebraic Formula Identity Problem (AFIT) is this: Given an algebraic formula with n indeterminates $x_1, \ldots, x_n$, addition and multiplication gates, and integer constants written in binary, does it compute the identically zero polynomial in $\mathbb{Z}[x_1, \ldots, x_n]$? It is well known that AFIT is in $\mathsf{co\text{-}RP}$, and hence in $\mathsf{P/poly}$. It should be intuitive that Buss' theory V^1_2 for exponential-time reasoning [Bus86] has the power to expand the given formula as an explicit sum of monomials with coefficients of polynomial bit complexity, and then carry over the induction on the number of variables in the proof of the Schwartz-Zippel Lemma. Is the full power of V^1_2 needed?

Problem 15. *Can the theory* T_2 *prove* $\mathsf{AFIT} \in \mathsf{P/poly}$*? Can Jerabek's theory* APC_1 *[Jer07] prove it? More precisely, prove (or disprove) there is an integer constant* $c \geq 1$ *such that the theory can prove the following sentence:*

$$\forall s{\in}Log\ \exists C{<}2^{s^c}\ \forall A{<}2^s\ (eval(C, A){=}1 \leftrightarrow \forall \bar{x}{<}2^{s^c}\ alg\text{-}eval(A, \bar{x}){=}0).$$

where $eval(C, A)$ *is the PV-symbol that represents the standard polynomial-time algorithm that evaluates the one-output Boolean circuit* C *on the binary representation of* A *as a string, and* $alg\text{-}eval(A, \bar{x})$ *is the PV-symbol that represents the standard polynomial-time algorithm that evaluates the algebraic formula* A *on the integer input* $\bar{x} = (x_1, \ldots, x_n)$.

Additional comments: Note that we are not asking the theory to prove that some (non-constructively given) non-uniform sequence of polynomial-size Boolean circuits for AFIT is correct; how would this even be stated in bounded arithmetic? We are only asking the theory to prove that such a sequence of Boolean circuits exists. This is not known to imply that AFIT is in P or in $\mathsf{NP} \cap \mathsf{co\text{-}NP}$ or any other such breakthrough.

The question is stated for AFIT but it could also be stated for finite fields, and for ACIT, the variant of the problem for algebraic circuits. In this last case, replace both occurrences of alg-eval$(A, \bar{x})$ by alg-eval-mod$(A, \bar{x}, m)$, where the latter represents the standard polynomial-time algorithm that evaluates the algebraic circuit A on $\bar{x}$ with arithmetic *modulo* m. Here, m is an s^c-bit integer that is to be quantified in the same way as $\bar{x}$.

Related work. The motivation for this question is to assess on the strength of $\mathsf{P/poly}$ in the theory V^0_2. See [ABM23].

References

[ABM23] Albert Atserias, Sam Buss, and Moritz Müller. "On the Consistency of Circuit Lower Bounds for Non-Deterministic Time". In: *CoRR* abs/2303.01016 (2023). To appear in the Proceedings of STOC 2023. arXiv: 2303.01016.

[Bus86] Samuel R. Buss. *Bounded Arithmetic.* Revision of 1985 Ph.D. Thesis (Department of Mathematics, Princeton University). Bibliopolis, 1986.

[Jer07] Emil Jerábek. "Approximate counting in bounded arithmetic". In: *J. Symb. Log.* 72.3 (2007), pp. 959–993.