Revisiting $\mathtt{MA}$ vs. $\exists \cdot \mathtt{BPP}$: A Functional Analogue

Takashi Ishizuka
Artificial Intelligence Laboratory, Fujitsu Limited, Japan
ishizuka-t@fujitsu.com

December 20, 2023

Abstract

The complexity class $\mathtt{TFNP}$, introduced by Megiddo and Papadimitriou [8], has played a crucial role in computational complexity theory and some real applied fields. In this short paper, we will revisit the definition of the class of search problems in $\mathtt{NP}$. Then, we formulate a probabilistic variant of $\mathtt{TFNP}$ (and also $\mathtt{FNP}$). Finally, we propose a functional analog of $\mathtt{MA}$ vs. $\exists \cdot \mathtt{BPP}$.

Keywords: Computational Complexity; Probabilistic Verification; Merlin-Arthur Model; Search Problems; TFNP

1 Appetizer

Some parts of this short paper have overlapped with Section 3.3.7 within the author's Ph.D. thesis [4]. We have answered a homework for the author left by that thesis. Thus, we will prove that $\mathtt{MA} = \exists \cdot \mathtt{BPP}$ if and only if $\mathtt{FMA} = \mathtt{gapFMA}$.

2 Backgrounds

Basic Notations We denote by $\mathbb{Z}$ and $\mathbb{R}$ the sets of all integers and real numbers, respectively. For an integer $a \in \mathbb{Z}$, we define $\mathbb{Z}_{\geq a} := \{x \in \mathbb{Z} : x \geq a\}$ and $\mathbb{Z}_{>a} := \{x \in \mathbb{Z} : x > a\}$. Similarly, for a real number $a \in \mathbb{R}$, we define $\mathbb{R}_{\geq a} := \{x \in \mathbb{Z} : x \geq a\}$ and $\mathbb{R}_{>a} := \{x \in \mathbb{Z} : x > a\}$. We write $[0, 1]$ for the interval of real numbers $\{x \in \mathbb{R} : 0 \leq x \leq 1\}$. Let X be a finite set. We denote by $|X|$ the cardinality of the elements in X.

Let $\{0, 1\}^*$ denote the set of binary strings with a finite length. For every string $x \in \{0, 1\}^*$, we denote by $|x|$ the length of x. For each positive integer n, we write $\{0, 1\}^n$ for the set of binary strength with the length n.

We write poly for the set of all non-zero polynomials with non-negative integer coefficients. We can straightforwardly see that every $f \in$ poly maps every positive integer to a positive integer. We denote by 1/poly the set of all functions that are inverse of a polynomial in poly, i.e., $1/\text{poly} := \{g : \mathbb{Z}_{>0} \to [0, 1] : \exists f \in \text{poly such that } g(n) = 1/f(n) \ \forall n \in \mathbb{Z}_{>0}\}$.

Let $a, b : \mathbb{Z}_{>0} \to [0, 1]$ be two polynomials. We say that a pair of polynomials (a, b) is *permissible* if a and b are polynomial-time computable and there is $q \in 1/\text{poly}$ such that $a(n) - b(n) \geq q(n)$ for every $n \in \mathbb{Z}_{>0}$.

Search Problems A total search problem [8] — the existence of solutions is guaranteed, and the correctness of every solution is efficiently checkable — comprises a fascinating field in computational complexity theory. Typically, the complexity class of search problems is formulated as follows.

Let $R \subseteq \{0, 1\}^* \times \{0, 1\}^*$ be a relation. We say that R is *polynomially balanced* if there is a polynomial $p \in$ poly such that for each $(x, y) \in R$, it holds that $|y| \leq p(|x|)$. We say that R is *polynomial-time decidable* if for each pair of strings $(x, y) \in \{0, 1\}^* \times \{0, 1\}^*$, we can decide whether $(x, y) \in R$ in polynomial time. We say that R is *total* if for every string $x \in \{0, 1\}^*$, there always exists at least one string y such that $(x, y) \in R$.

For a relation $R \subseteq \{0, 1\}^* \times \{0, 1\}^*$, the search problem with respect to R is defined as follows: Given a string $x \in \{0, 1\}^*$, find a string $y \in \{0, 1\}^*$ such that $(x, y) \in R$ if such a y exists, otherwise reports "*no*." When R is also total, we call such a search problem a total search problem. The complexity class $\mathtt{FNP}$ is the set of all search problems with respect to a polynomially balanced and polynomial-time decidable relation R. The complexity class $\mathtt{TFNP}$ is the set of all total search problems belonging to $\mathtt{FNP}$. By definition, it holds that $\mathtt{TFNP} \subseteq \mathtt{FNP}$.

Probabilistic Variants of FNP In this paper, we revisit the notion of polynomial-time decidable. Naturally, it means decidable by a deterministic algorithm. We extend this definition to a probabilistic verification procedure.

Roughly speaking, a probabilistic verification procedure is a family of polynomial-time uniform Boolean circuits $V = \{V_n : n \in \mathbb{Z}_{>0}\}$ with V_n taking as input (x, z) where $x \in \{0,1\}^n$ is a binary string of length n, z is a random string of length $r(|x|)$, and r is some polynomial in poly.

Let $R \subseteq \{0,1\}^* \times \{0,1\}^*$ be a polynomially balanced relation. We say that R is polynomial-time probabilistic decidable if there is a probabilistic verification procedure $V = \{V_n : n \in \mathbb{Z}_{>0}\}$ such that for every pair of strings $(x, y) \in \{0,1\}^* \times \{0,1\}^*$, V decides whether $(x, y) \in R$ with high probability.

The above idea leads us to a probabilistic extension of the class FNP. This short paper investigates the properties of such a complexity class.

Related Works Massar and Santha [7, 6] have studied a quantum counterpart of FNP. They introduced the complexity class FQMA and TFQMA and have shown some quantum total search problems belonging to TFQMA. Our definition is inspired by their definition. In contrast to [7, 6], this paper focuses on the classical Merlin-Arthur model.

The complexity class MA, introduced by Babai [1], is a set of all decision problems that are solvable by *Merlin-Arthur protocol*, in which Merlin who has unbounded computational power sends a polynomial-size message to Arthur whose computational power is probabilistic polynomial-time, and then, Arthur must verify whether the message is correct with high probability. More formally, every problem A in MA has a polynomial $p \in$ poly and a probabilistic verification procedure $V := \{V_n^r : n \in \mathbb{Z}_{>0}\}$ such that for every input string x of length n, (i) if the answer is "*yes*," then there is a witness $y \in \{0,1\}^{p(n)}$ such that $\Pr[V_{n+p(n)}^r(x\sharp y) = 1] \geq 2/3$; otherwise (ii) if the answer is "*no*," then for every string $y \in \{0,1\}^{p(n)}$, $\Pr[V_{n+p(n)}^r(x\sharp y) = 1] \leq 1/3$.

It is known that there is a very similar class to MA called $\exists \cdot$ BPP[1]. For every problem A in $\exists \cdot$ BPP and for each input x, its witness space is separated into two subspaces $\hat{A}(x)$ and $\check{A}(x)$ such that every string $y \in \hat{A}(x)$ is accepted with high probability and every witness string $y \in \hat{A}(x)$ is rejected with high probability. More formally, every problem $A = (\hat{A}, \check{A})$ in $\exists \cdot$ BPP has a polynomial p in poly and a probabilistic verification procedure $V := \{V_n^r : n \in \mathbb{Z}_{>0}\}$ such that for every input string x of length n, (i) if the answer is "*yes*," then there is a string $y \in \hat{A}(x)$ such that $\Pr[V_{n+p(n)}^r(x\sharp y) = 1] \geq 2/3$; otherwise (ii) if the answer is "*no*," then for every string $y \in \check{A}(x)$, $\Pr[V_{n+p(n)}^r(x\sharp y) = 1] \leq 1/3$, where $\hat{A}(x) \cap \check{A}(x) = \emptyset$ and $\hat{A}(x) \cup \check{A}(x) = \{0,1\}^{p(n)}$.

These two classes, MA and $\exists \cdot$ BPP, seem to be the same, but Fenner et al. [2] have shown an oracle separation between MA and $\exists \cdot$ BPP. In this short paper, we propose a functional counterpart of MA vs. $\exists \cdot$ BPP.

3 Functional Merlin-Arthur Model

In this section, we formally define the probabilistic procedure and some classes related to the probabilistic procedures.

Let r be a polynomial in poly. A probabilistic verification procedure for the input length $n \in \mathbb{Z}_{>0}$ is a family of polynomial-time uniform Boolean circuits

$$V := \left\{ V_n^{p,r} : \{0,1\}^n \times \{0,1\}^{p(n)} \times \{0,1\}^{r(n)} \to \{0,1\} \right\}_{n \in \mathbb{Z}_{>0}},$$

where p and r are fixed polynomials in poly, and for each positive integer n, the Boolean circuit $V_n^{p,r}$ has n-bits input register, $p(n)$-bits witness register, $r(n)$-bits random register, and 1-bit output register. For each input string $x \in \{0,1\}^n$ and each witness $y \in \{0,1\}^{p(n)}$, the acceptance probability of (x, y) by the procedure $V_n^{p,r}$ is defined as $|\{z : V_n^{p,r}(x, y, z) = 1\}|/2^{r(n)}$. We write $\Pr[V_n^{p,r}(x, y) = 1]$ for the acceptance probability of (x, y) by the procedure V_n^r.

For a relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$, we use $R(x) := \{y \in \{0,1\}^* : (x, y) \in R\}$ for every string $x \in \{0,1\}^*$. To deal with computational problems that have probabilistic features, we focus on *promise* problems. A promise problem A is a pair of two relations $\hat{A}, \check{A} \subseteq \{0,1\}^* \times \{0,1\}^*$ satisfying that for every string $x \in \{0,1\}^*$, $\hat{A}(x) \cap \check{A}(x) = \emptyset$ and $\hat{A}(x) \cup \check{A}(x) \neq \emptyset$. Here, we call $\hat{A}$ and $\check{A}$ a *accepting subspace* and a *rejecting subspace*, respectively. A promise problem $A = (\hat{A}, \check{A})$ is said to be polynomially balanced if the relations $\hat{A}$ and $\check{A}$ are polynomially balanced.

Let (a, b) be a permissible pair of polynomials, and let $A = (\hat{A}, \check{A})$ denote a polynomially balanced promise problem. We say that A is (a, b)-verifiable if there is a probabilistic verification procedure $V := \{V_n^{p,r} : n \in \mathbb{Z}_{>0}\}$ such that for every input string x of length n, exactly one of the following holds:

[1] See also Complexity Zoo: BPP with Existence Operator.

(i) $\hat{A}(x) \neq \emptyset$ and for every $y \in \hat{A}(x)$, $\Pr[V_n^{p,r}(x,y) = 1] \geq a(n)$;

(ii) for every $y \in \{0,1\}^{p(n)} = \check{A}(x)$, $\Pr[V_n^{p,r}(x,y) = 1] \leq b(n)$.

In the rest of this short paper, we simply call a promise problem that is polynomially balanced and (a,b)-verifiable a polynomial-time (a,b)-verifiable problem. Furthermore, we omit (a,b) when the permissible pair of polynomials (a,b) is clear from the context.

We are now ready to define the complexity class, which is a functional counterpart of $\mathtt{MA}(a,b)$. Let (a,b) be a permissible pair of polynomials, and let $A = (\hat{A}, \check{A})$ denote a polynomial-time (a,b)-verifiable problem. A search problem with respect to A is formulated as: Given an input $x \in \{0,1\}^*$, find $y \in \hat{A}(x)$ if $\hat{A} \neq \emptyset$; otherwise report "*no*." The complexity class $\mathtt{FMA}(a,b)$ is the set of all search problems with respect to polynomial-time (a,b)-verifiable problems. The complexity class $\mathtt{TFMA}(a,b)$ is the subset of $\mathtt{FMA}(a,b)$ satisfying that $R^{\geq a}(x) \neq \emptyset$ for every input string $x \in \{0,1\}^*$. We simply write $\mathtt{FMA}$ and $\mathtt{TFMA}$ for $\mathtt{FMA}(2/3,1/3)$ and $\mathtt{TFMA}(2/3,1/3)$, respectively.

3.1 Reduction in Search Problems

This section defines a polynomial-time reduction between two search problems.

Let (a,b) be a permissible pair of polynomials. Let $R = \left(\hat{R}^{\geq a}, \check{R}^{\leq b}\right)$ be a (a,b)-verifiable problem, and let $S = \left(\hat{S}^{\geq a'}, \check{S}^{\leq b'}\right)$ be a (a',b')-verifiable problem. A polynomial-time reduction from R to S is defined by two polynomial-time computable functions $f : \{0,1\}^* \to \{0,1\}^*$ and $g : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ satisfying the following properties:

1. For every input x of R, the string $f(x)$ is an input of S.
2. If $\hat{R}^{\geq a}(x) \neq \emptyset$, then it holds that $\hat{S}^{\geq a'}(f(x)) \neq \emptyset$ and $g(x,y) \in \hat{R}^{\geq a}(x)$ for every $y \in \hat{S}^{\geq a'}(x)$.
3. If $\hat{R}^{\geq a}(x) \neq \emptyset$, then it holds that either $\hat{S}^{\geq a'}(f(x)) \neq \emptyset$ or for every $y \in \hat{S}^{\geq a'}(f(x))$, the string $g(x,y)$ is a succinct witness to show $\check{R}^{\geq a}(x) = \emptyset$.

3.2 Properties of FMA

This section observes some properties of the class $\mathtt{FMA}$. We first show that the class $\mathtt{FMA}$ achieves perfect completeness in the same way as the decision class $\mathtt{MA}$.

Proposition 1. *Let (a,b) be a permissible pair of polynomials. It holds that* $\mathtt{FMA}(a,b) = \mathtt{FMA}(1,1/2)$.

Proof. Apply the same technique provided by [3, 9] to show the perfect completeness of the Merlin-Arthur protocol. □

Next, we focus on the extremal soundness of the class $\mathtt{FMA}$. We write $\mathtt{FMA}(1,<1)$ for the class of all polynomially $(1,<1)$-verifiable problems $R = \left(\hat{R}^{\geq 1}, \check{R}^{<1}\right)$, where for every string $x \in \{0,1\}^*$, the set $R^{<1}(x)$ is the set of all strings with non-zero probability of rejection.

We now show that the class $\mathtt{FMA}(1,<1)$ is equal to the complexity class $\mathtt{TF}\Sigma_2^{\mathtt{P}}$, introduced by Kleinberg et al. [5]. The class $\mathtt{TF}\Sigma_2^{\mathtt{P}}$ is the set of all relations $R \subseteq \{0,1\}^* \times \{0,1\}^*$ satisfying that there are polynomials p and r in poly such that (i) for every $x \in \{0,1\}^*$, there is a string $y \in \{0,1\}^{p(|x|)}$ such that $(x,y) \in R$; and (ii) there is a family of polynomial-time uniform Boolean circuits $M = \{M_n^{p,r} : \{0,1\}^n \times \{0,1\}^{p(n)} \times \{0,1\}^{r(n)} \to \{0,1\}\}_{n \in \mathbb{Z}_{>0}}$ such that $(x,y) \in R$ if and only if $M_{|x|}^{p,r}(x,y,z) = 1$ for every $z \in \{0,1\}^{r(|x|)}$. Note that every relation R in $\mathtt{TF}\Sigma_2^{\mathtt{P}}$ is a polynomially-balanced total relation by definition.

Proposition 2. $\mathtt{FMA}(1,<1) = \mathtt{TF}\Sigma_2^{\mathtt{P}}$

Proof. It is not hard to see that $\mathtt{FMA}(1,<1) \subseteq \mathtt{TF}\Sigma_2^{\mathtt{P}}$. Therefore, we prove the reverse containment: $\mathtt{TF}\Sigma_2^{\mathtt{P}} \subseteq \mathtt{FMA}(1,<1)$.

Let $A \subseteq \{0,1\}^* \times \{0,1\}^*$ be any polynomially balanced relation in $\mathtt{TF}\Sigma_2^{\mathtt{P}}$. We denote by $M = \{M_n^{p,r} : n \in \mathbb{Z}_{>0}\}$ a polynomial-time uniform Boolean circuits corresponding to A. Then, there is a naturally induced promise problem $A = (\hat{A}, \check{A})$ such that for every string $x \in \{0,1\}^*$,

$$\hat{A}(x) := \{y \in \{0,1\}^{p(|x|)} : M_{|x|}^{p,r}(x,y,z) = 1 \forall z \in \{0,1\}^{r(|x|)}\}$$

and

$$\check{A}(x) := \{y \in \{0,1\}^{p(|x|)} : \exists z \in \{0,1\}^{r(|x|)} \text{ s. t. } M_{|x|}^{p,r}(x,y,z) = 0\}.$$

Hence, we can easily see that the promise problem $A = (\hat{A}, \check{A})$ belongs to $\mathtt{FMA}(1,<1)$. □

4 Functional Counterpart of $\texttt{MA}$ vs. $\exists \cdot \texttt{BPP}$

Recall that, in the class $\texttt{FMA}$, we only consider the *promise* problems. Let $R = \left(\hat{R}^{\geq a}, \check{R}^{\leq b}\right)$ be a polynomially (a, b)-verifiable problem. It is straightforward to see that for some polynomial p, it holds that $\hat{R}^{\geq a}(x) \cup \check{R}^{\leq b}(x) \subseteq \{0,1\}^{p(|x|)}$ for every input string x. Note that, in our definition, there may exist a string $y \in \{0,1\}^{p(|x|)} \setminus \left(\hat{R}^{\geq a}(x) \cup \check{R}^{\leq b}(x)\right)$ when the acceptance subspace $\hat{R}^{\geq a}(x)$ is non-empty.

We now consider the new variant of $\texttt{FMA}$ which is promised that $R^{\geq a}(x) \cup R^{\leq b}(x) = \{0,1\}^{p(|x|)}$ for every input string x. We call this class $\texttt{gapFMA}(a, b)$. Also, we simply write $\texttt{gapFMA}$ for $\texttt{gapFMA}(2/3, 1/3)$.

It is not hard to see that $\texttt{gapFMA}(a, b) \subseteq \texttt{FMA}(a, b)$. At this moment, a natural and interesting question arises: Does $\texttt{gapFMA}$ contain $\texttt{FMA}$? Unfortunately, it would be abstruse to resolve. As shown in Theorem 3, we can regard the class $\texttt{gapFMA}$ as a functional counterpart of $\exists \cdot \texttt{BPP}$.

Theorem 3. $\texttt{MA} = \exists \cdot \texttt{BPP}$ *if and only if* $\texttt{FMA} = \texttt{gapFMA}$

Proof. $(\Longrightarrow)$ It is trivial.

$(\Longleftarrow)$ From the assumption, for every polynomially (a, b)-verifiable problem $R = (\hat{R}, \check{R}) \in \texttt{FMA}$, we have a reduction from L to another problem $S \in \texttt{gapFMA}$. By definition, it holds that $\hat{S}(x) \cup \check{S}(x) = \{0,1\}^{p(|x|)}$ for every input string x. Furthermore, we have a witness $g(x, y) \in \hat{R}(x)$ for each witness $y \in \hat{S}(f(x))$. Therefore, we can see that $S \in \exists \cdot \texttt{BPP}$ when we regard S as a decision problem. □

5 Conclusion

In this short paper, we have investigated the theoretical features of a probabilistic analog of $\texttt{FNP}$. In particular, we have proposed a functional counterpart of $\texttt{MA}$ vs. $\exists \cdot \texttt{BPP}$.

From Proposition 1, it seems that $\texttt{FNP}$ and $\texttt{FMA}$ are the same. However, Theorem 3 implies that there might be a gap between these two classes. Note that from the simple observation, we obtain the following two equations: $\texttt{FNP} = \texttt{gapFMA}(1, 0)$ and $\texttt{TFNP} = \texttt{gapTFMA}(1, 0)$. Our revisit in this short paper implies that it is important to shed light on the rejecting subspaces.

Acknowledgment

This work was supported by JST, ACT-X, Grant Number JPMJAX2101. I have also been grateful to my supporters (the supervisor, friends, families, and so on) during my college life.

References

[1] László Babai. "Trading Group Theory for Randomness." In: *Proceedings of the 17th Annual ACM Symposium on Theory of Computing.* ACM, 1985, pp. 421–429. DOI: 10.1145/22145.22192.

[2] Stephen A. Fenner, Lance Fortnow, Stuart A. Kurtz, and Lide Li. "An oracle builder's toolkit." In: *Inf. Comput.* 182.2 (2003), pp. 95–136. DOI: 10.1016/S0890-5401(03)00018-X.

[3] Oded Goldreich and David Zuckerman. "Another Proof That BPP $\subseteq$ PH (and More)." In: *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation.* Vol. 6650. Lecture Notes in Computer Science. Springer, 2011, pp. 40–53. DOI: 10.1007/978-3-642-22670-0_6.

[4] Takashi Ishizuka. "On TFNP Classes: Approaches from Fixed Point Theory and Algorithmic Game Theory." In: (2022).

[5] Robert Kleinberg, Oliver Korten, Daniel Mitropolsky, and Christos H. Papadimitriou. "Total Functions in the Polynomial Hierarchy." In: *12th Innovations in Theoretical Computer Science Conference.* Vol. 185. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, 44:1–44:18. DOI: 10.4230/LIPICS.ITCS.2021.44.

[6] Serge Massar and Miklos Santha. "Characterising the intersection of QMA and coQMA." In: *Quantum Inf. Process.* 20.12 (2021), p. 396. DOI: 10.1007/S11128-021-03326-3.

[7] Serge Massar and Miklos Santha. "Total functions in QMA." In: *Quantum Inf. Process.* 20.1 (2021), p. 35. DOI: 10.1007/S11128-020-02959-0.

[8] Nimrod Megiddo and Christos H. Papadimitriou. "On Total Functions, Existence Theorems and Computational Complexity." In: *Theoretical Computer Science* 81.2 (1991), pp. 317–324. DOI: 10.1016/0304-3975(91)90200-L.

[9] Stathis Zachos and Martin Fürer. "Probabalistic Quantifiers vs. Distrustful Adversaries." In: *Foundations of Software Technology and Theoretical Computer Science*. Vol. 287. Lecture Notes in Computer Science. Springer, 1987, pp. 443–455. DOI: 10.1007/3-540-18625-5_67.