Theory of Computing Systems (2023) 67:383–412
https://doi.org/10.1007/s00224-022-10105-9

Complexity Limitations on One-turn Quantum Refereed Games

Soumik Ghosh[1] · **John Watrous**[2]

Accepted: 30 September 2022 / Published online: 10 December 2022

Abstract
This paper studies complexity theoretic aspects of quantum refereed games, which are abstract games between two competing players that send quantum states to a referee, who performs an efficiently implementable joint measurement on the two states to determine which of the player wins. The complexity class QRG(1) contains those decision problems for which one of the players can always win with high probability on yes-instances and the other player can always win with high probability on no-instances, regardless of the opposing player's strategy. This class trivially contains QMA ∪ co-QMA and is known to be contained in PSPACE. We prove stronger containments on two restricted variants of this class. Specifically, if one of the players is limited to sending a classical (probabilistic) state rather than a quantum state, the resulting complexity class CQRG(1) is contained in $\exists \cdot \mathrm{PP}$ (the nondeterministic polynomial-time operator applied to PP); while if both players send quantum states but the referee is forced to measure one of the states first, and incorporates the classical outcome of this measurement into a measurement of the second state, the resulting class MQRG(1) is contained in $\mathrm{P} \cdot \mathrm{PP}$ (the unbounded-error probabilistic polynomial-time operator applied to PP).

Keywords Quantum computing · Complexity theory · Game theory

1 Introduction

Abstract notions of games have long played an important role in complexity theory. For example, combinatorial games provide complete problems for various complexity classes [14], the notion of alternation is naturally described in game-theoretic

✉ Soumik Ghosh
soumikghosh@uchicago.edu

[1] Department of Computer Science, University of Chicago, Chicago, USA

[2] Institute for Quantum Computing and School of Computer Science, University of Waterloo, Waterloo, Canada

terms [12], and interactive proof systems [3, 5, 22, 23] and many variants of them are naturally formulated as games [13, 18].

This paper is concerned with games between two competing, computationally unbounded players, administered by a computationally bounded referee. In the classical setting, complexity theoretic aspects of games of this form were investigated in the 1990s by Koller and Megiddo [29], Feigenbaum, Koller, and Shor [18], Condon, Feigenbaum, Lund, and Shor [10, 11], and Feige and Kilian [17]. Quantum computational analogues of these games were later considered in [24–26], and [28].

Our focus will be on *one-turn refereed games*, in which the players and the referee first receive a common input string, and then each player sends a single polynomial-length (quantum or classical) message to the referee, who then decides which player has won. We will refer to the two competing players as *Alice* and *Bob* for convenience. In the classical case Alice and Bob's messages may in general be described by probability distributions over strings, while in the quantum case Alice and Bob's messages are described by mixed quantum states, which are represented by density operators. In both cases, the referee's decision process must be specified by a polynomial-time generated family of (quantum or classical) circuits. Two complexity classes are defined—RG(1) in the classical setting[1] and QRG(1) in the quantum setting—consisting of all promise problems $A = (A_{\text{yes}}, A_{\text{no}})$ for which there exists a game (either classical or quantum, respectively) such that Alice can win with high probability on inputs $x \in A_{\text{yes}}$ and Bob can win with high probability on inputs $x \in A_{\text{no}}$, regardless of the other player's behavior.

In essence, the complexity classes RG(1) and QRG(1) may be viewed as extensions of the classes MA and QMA in which *two competing Merlins*, one whose aim is to convince the referee (whose role is analogous to Arthur, also called the verifier, in the case of MA and QMA) that the input string is a yes-instance of a given problem, and the other whose aim is to convince the referee that the input string is a no-instance.

It is known that the complexity class RG(1) is equal to S_2^p, which refers to the second level of the *symmetric polynomial-time hierarchy* introduced by Canetti [9] and Russell and Sundaram [35]. This class is most typically defined in terms of quantifiers that suggest games in which Alice and Bob choose polynomial-length strings (as opposed to probability distributions of strings) to send to the referee, but the class does not change if one adopts a bounded-error definition in which Alice and Bob are allowed to make use of randomness [16]. Moreover, the class does not change if the referee is permitted the use of randomness, again assuming a bounded-error definition. An essential fact through which these equivalences may be proved, due to Althöfer [2] and Lipton and Young [32], is that non-interactive randomized games always admit near-optimal strategies that are uniform over polynomial-size sets of strings. It is also known that RG(1) is closed under Cook reductions [35] and satisfies $\mathrm{RG}(1) \subseteq \mathrm{ZPP}^{\mathrm{NP}}$ [8].

[1] We note explicitly that this nomenclature clashes with [17], which defines RG(1) in terms of *one-round* (i.e., two-turn) refereed games, which is RG(2) with respect to our naming conventions.

In contrast to the containment $\mathrm{RG}(1) \subseteq \mathrm{ZPP}^{\mathrm{NP}}$, the best upper-bound known for QRG(1) is that this class is contained in PSPACE [28]. It is reasonable to conjecture that a stronger upper-bound on QRG(1) can be proved.

Indeed, Gutoski and Wu [27] proved that even the class QRG(2), which is analogous to QRG(1) except that the referee first sends a message to Alice and Bob and then receives responses from them, is contained in PSPACE. The two classes are in fact equal, meaning $\mathrm{QRG}(2) = \mathrm{PSPACE}$, as a consequence of the trivial containment $\mathrm{RG}(2) \subseteq \mathrm{QRG}(2)$ together with the known equality $\mathrm{RG}(2) = \mathrm{PSPACE}$ [17]. While not directly relevant to our results, we note that the classes $\mathrm{RG} = \mathrm{RG}(\mathrm{poly})$ and $\mathrm{QRG} = \mathrm{QRG}(\mathrm{poly})$ defined in an analogous way, but allowing any polynomial number of messages between the referee and Alice and Bob are both equal to EXP [17, 26].

In this work we consider two restricted variants of QRG(1), and prove stronger upper-bounds than PSPACE on these restricted variants. The first variant is one in which Alice is limited to sending a classical message to the referee, while Bob is free to send a quantum state. The resulting class, which we call CQRG(1), is proved to be contained in $\exists \cdot \mathrm{PP}$ (the class obtained when the nondeterministic polynomial-time operator is applied to PP). This containment follows from an application of the Althöfer–Lipton–Young technique mentioned above, although in the quantum setting the proof requires relatively recent tail bounds on sums of matrix-valued random variables, as opposed to a more standard Hoeffding–Chernoff type of bound that suffices in the classical case. In particular, we make use of a tail bound of this sort due to Tropp [36]. The second variant we consider is one in which both Alice and Bob are free to send quantum states, but where the referee must first measure Alice's state and then incorporate the classical outcome of this measurement into a measurement of Bob's state. We call the corresponding class MQRG(1), and prove the containment $\mathrm{MQRG}(1) \subseteq \mathrm{P} \cdot \mathrm{PP}$ (the class obtained when the unbounded error probabilistic polynomial-time operator is applied to PP). Note that $\exists \cdot \mathrm{PP}$ is contained in $\mathrm{P} \cdot \mathrm{PP}$, which is, in turn, contained in PSPACE.

2 Preliminaries

We assume the reader is familiar with basic aspects of computational complexity theory and quantum information and computation. There are four subsections included in this preliminaries section, the first of which clarifies a few specific concepts, conventions, and definitions concerning complexity theory. The second subsection is concerned specifically with counting complexity, and presents a development of some results on this topic that are central to this paper. Proofs are included because these results represent minor generalizations of known results on counting complexity. The third subsection discusses a few specific definitions and concepts from quantum information and computation, along with a proof of a fact that may be considered a known result, but for which a complete proof does not appear in published form. The final subsection states the tail bound due to Tropp mentioned above.

2.1 Complexity theory basics

Throughout this paper, languages, promise problems, and functions on strings are assumed to be over the binary alphabet $\Sigma = \{0, 1\}$. The set of natural numbers, including 0, is denoted $\mathbb{N}$.

A function of the form $p : \mathbb{N} \to \mathbb{N}$ is said to be *polynomially bounded* if there exists a deterministic Turing machine that runs in polynomial time and outputs $0^{p(n)}$ on input 0^n for all $n \in \mathbb{N}$. Unless it is explicitly indicated otherwise, the input of a given polynomially bounded function p is assumed to be the natural number $|x|$, for whatever input string $x \in \Sigma^*$ is being considered at that moment. With this understanding in mind, we will write p in place of $p(|x|)$ when referring to the natural number output that is determined in this way. For example, in Definition 1 below, all of the occurrences of p in the displayed equations are short for $p(|x|)$. This convention helps to make formulas and equations more clear and less cluttered.

A promise problem is a pair $A = (A_{\text{yes}}, A_{\text{no}})$ of sets of strings $A_{\text{yes}}, A_{\text{no}} \subseteq \Sigma^*$ with $A_{\text{yes}} \cap A_{\text{no}} = \varnothing$. Strings in A_{yes} represent yes-instances of a decision problem, strings in A_{no} represent no-instances, and all other strings represent "don't care" inputs for which no restrictions are placed on a hypothetical computation for that problem.

We fix a pairing function that efficiently encodes two strings $x, y \in \Sigma^*$ into a single binary string denoted $\langle x, y \rangle \in \Sigma^*$, and we assume that this function satisfies some simple properties:

1. The length of the pair $\langle x, y \rangle$ depends only on the lengths $|x|$ and $|y|$, and is polynomial in these lengths.
2. The computation of x and y from $\langle x, y \rangle$, as well as the computation of $\langle x, y \rangle$ from x and y, can be performed deterministically in polynomial time.

One suitable choice for such a function is suggested by the equation

$$\langle a_1 a_2 \cdots a_n, b_1 b_2 \cdots b_m \rangle = 0a_1 0a_2 \cdots 0a_n 1 b_1 b_2 \cdots b_m \tag{1}$$

for $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m \in \Sigma$. Any such pairing function may be extended recursively to obtain a tuple function for any fixed number of inputs by taking

$$\langle x_1, x_2, x_3, \ldots, x_k \rangle = \langle \langle x_1, x_2 \rangle, x_3, \ldots, x_k \rangle \tag{2}$$

for strings $x_1, \ldots, x_k \in \Sigma^*$, where $k \geq 3$. Hereafter, when we refer to the computation of any function taking multiple string-valued arguments, we assume that these input strings have been encoded into a single string using this tuple function. For instance, when f is a function that represents a computation, we write $f(x, y, z)$ rather than $f(\langle x, y, z \rangle)$.

Finally, we define the nondeterministic and probabilistic polynomial-time operators, which may be applied to an arbitrary complexity class, as follows.

Definition 1 For a given complexity class of languages $\mathcal{C}$, the complexity classes $\exists \cdot \mathcal{C}$ and $\mathrm{P} \cdot \mathcal{C}$ are defined as follows.

1. The complexity class $\exists \cdot \mathcal{C}$ contains all promise problems $A = (A_{\text{yes}}, A_{\text{no}})$ for which there exists a language $B \in \mathcal{C}$ and a polynomially bounded function p such that these two implications hold:
$$\begin{aligned} x \in A_{\text{yes}} &\Rightarrow \left\{ y \in \Sigma^p \,:\, \langle x, y \rangle \in B \right\} \neq \varnothing, \\ x \in A_{\text{no}} &\Rightarrow \left\{ y \in \Sigma^p \,:\, \langle x, y \rangle \in B \right\} = \varnothing. \end{aligned} \tag{3}$$
2. The complexity class $\mathrm{P} \cdot \mathcal{C}$ contains all promise problems $A = (A_{\text{yes}}, A_{\text{no}})$ for which there exists a language $B \in \mathcal{C}$ and a polynomially bounded function p such that these two implications hold:
$$\begin{aligned} x \in A_{\text{yes}} &\Rightarrow \left| \left\{ y \in \Sigma^p \,:\, \langle x, y \rangle \in B \right\} \right| > \frac{1}{2} \cdot 2^p, \\ x \in A_{\text{no}} &\Rightarrow \left| \left\{ y \in \Sigma^p \,:\, \langle x, y \rangle \in B \right\} \right| \leq \frac{1}{2} \cdot 2^p. \end{aligned} \tag{4}$$

2.2 Counting complexity

Counting complexity is principally concerned with the number of solutions to certain computational problems. Readers interested in learning more about counting complexity and some of its applications are referred to the survey paper of Fortnow [19]. As was suggested at the beginning of the current section, we will require some basic results on counting complexity that represent minor generalizations of known results. We begin with the following definition.

Definition 2 Let $\mathcal{C}$ be any complexity class of languages over the alphabet Σ. A function $f : \Sigma^* \to \mathbb{Z}$ is a $\mathrm{Gap} \cdot \mathcal{C}$ function if there exist languages $A, B \in \mathcal{C}$ and a polynomially bounded function p such that
$$f(x) = \left| \left\{ y \in \Sigma^p \,:\, \langle x, y \rangle \in A \right\} \right| - \left| \left\{ y \in \Sigma^p \,:\, \langle x, y \rangle \in B \right\} \right| \tag{5}$$
for all $x \in \Sigma^*$.

We observe that this definition is slightly non-standard, as gap functions are usually defined in terms of differences between the number of accepting and rejecting computations of nondeterministic machines (as opposed to a difference involving two potentially unrelated languages A and B). It is also typical that one focuses on specific choices for $\mathcal{C}$, particularly $\mathcal{C} = \mathrm{P}$. Our definition is, however, equivalent to the traditional definition in this case, and we will write GapP rather than $\mathrm{Gap} \cdot \mathrm{P}$ so as to be consistent with the standard name for this class of functions. We will also be interested in the case $\mathcal{C} = \mathrm{PP}$, which yields a class of functions $\mathrm{Gap} \cdot \mathrm{PP}$ that is less commonly considered.

A key feature of the class of GapP functions that facilitates its use is that it possesses strong closure properties. This is true more generally for the class $\mathrm{Gap} \cdot \mathcal{C}$ provided that $\mathcal{C}$ itself possesses certain properties. For the closure properties we

require, it suffices that $\mathcal{C}$ is nontrivial (meaning that $\mathcal{C}$ contains at least one language that is not equal to $\varnothing$ or Σ^*) and is closed under the join operation as well as polynomial-time truth-table reductions. (The *join* of languages A and B is defined as $\{x0 : x \in A\} \cup \{x1 : x \in B\}$.) These properties are, of course, possessed by both P and PP, with the closure of PP under truth table reductions having been proved by Fortnow and Reingold [20] based on methods developed by Beigel, Reingold, and Spielman [6].

The following proposition is immediate from the definitions of $\text{Gap} \cdot \mathcal{C}$ and $\text{P} \cdot \mathcal{C}$.

Proposition 3 *Let $\mathcal{C}$ be a complexity class of languages that is closed under complementation and joins. A promise problem $A = (A_{yes}, A_{no})$ is contained in $\text{P} \cdot \mathcal{C}$ if and only if there exists a $\text{Gap} \cdot \mathcal{C}$ function f such that*

$$\begin{aligned} x \in A_{yes} &\Rightarrow f(x) > 0, \\ x \in A_{no} &\Rightarrow f(x) \leq 0. \end{aligned} \tag{6}$$

The lemmas that follow establish the specific closure properties we require. For the first property the assumption that $\mathcal{C}$ is closed under joins and polynomial-time truth-table reductions is not required; closure under Karp reductions suffices.

Lemma 4 *Let $\mathcal{C}$ be a nontrivial complexity class of languages that is closed under Karp reductions. Let $f \in \text{Gap} \cdot \mathcal{C}$ and let p be a polynomially bounded function. The function*

$$g(x) = \sum_{y \in \Sigma^p} f(x, y) \tag{7}$$

is a $\text{Gap} \cdot \mathcal{C}$ function.

Proof By the assumption that $f \in \text{Gap} \cdot \mathcal{C}$, there exists a polynomially bounded function q and languages $A_0, A_1 \in \mathcal{C}$ such that

$$f(x, y) = \left|\left\{z \in \Sigma^{q(|\langle x,y\rangle|)} : \langle x, y, z\rangle \in A_0\right\}\right| - \left|\left\{z \in \Sigma^{q(|\langle x,y\rangle|)} : \langle x, y, z\rangle \in A_1\right\}\right| \tag{8}$$

for all $x \in \Sigma^*$ and $y \in \Sigma^p$. By the assumptions on our pairing function described above, it is the case that $|\langle x, y\rangle|$ depends only on $|x|$ and $|y|$, and therefore there exists a (necessarily polynomially bounded) function r such that $r(|x|) = p(|x|) + q(|\langle x, y\rangle|)$ for all $x \in \Sigma^*$ and $y \in \Sigma^p$. Define

$$\begin{aligned} B_0 &= \left\{\langle x, yz\rangle : y \in \Sigma^p,\ z \in \Sigma^{q(|\langle x,y\rangle|)},\ \langle x, y, z\rangle \in A_0\right\}, \\ B_1 &= \left\{(x, yz) : y \in \Sigma^p,\ z \in \Sigma^{q(|\langle x,y\rangle|)},\ \langle x, y, z\rangle \in A_1\right\}. \end{aligned} \tag{9}$$

By the nontriviality and closure of $\mathcal{C}$ under Karp reductions, it is evident that $B_0, B_1 \in \mathcal{C}$. It may be verified that

$$g(x) = \left|\left\{w \in \Sigma^r : \langle x, w\rangle \in B_0\right\}\right| - \left|\left\{w \in \Sigma^r : \langle x, w\rangle \in B_1\right\}\right| \tag{10}$$

for all $x \in \Sigma^*$, and therefore $g \in \text{Gap} \cdot \mathcal{C}$. □

For the next lemma, and elsewhere in the paper, we will use the following notation for convenience: Σ_1^n denotes the set of all strings over the binary alphabet Σ that have length n and contain exactly one occurrence of the symbol 1. It is therefore the case that $|\Sigma_1^n| = n$.

Lemma 5 *Let $\mathcal{C}$ be a nontrivial complexity class of languages that is closed under joins and polynomial-time truth table reductions. Let $f \in Gap \cdot \mathcal{C}$ and let p be a polynomially bounded function. The function*

$$g(x) = \prod_{y \in \Sigma_1^p} f(x, y) \tag{11}$$

is a $Gap \cdot \mathcal{C}$ function.

Proof Given that $f \in \text{Gap} \cdot \mathcal{C}$, there exists a polynomially bounded function q and languages $A_0, A_1 \in \mathcal{C}$ such that

$$f(x, y) = \left|\left\{z \in \Sigma^{q(|(x,y)|)} \,:\, \langle x, y, z\rangle \in A_0\right\}\right| - \left|\left\{z \in \Sigma^{q(|(x,y)|)} \,:\, \langle x, y, z\rangle \in A_1\right\}\right| \tag{12}$$

for all $x, y \in \Sigma^*$. We may assume further that A_0 and A_1 are disjoint languages, for if they are not, we may replace A_0 and A_1 with $A_0 \cap \overline{A_1}$ and $A_1 \cap \overline{A_0}$, respectively; this does not change the value of the right-hand side of the (12), and the languages $A_0 \cap \overline{A_1}$ and $A_1 \cap \overline{A_0}$ must both be contained in $\mathcal{C}$ for $A_0, A_1 \in \mathcal{C}$ by the closure of $\mathcal{C}$ under joins and truth-table reductions.

By the assumptions on our pairing function described above, there exists a polynomially bounded function r such that $r(|x|) = q(|(x, y)|)$ for all $x \in \Sigma^*$ and $y \in \Sigma^p$. We will write $y_1, \ldots, y_p$ to denote the elements of Σ_1^p sorted in lexicographic order. Define two languages B_0 and B_1 as follows:

- B_0 is the language of all pairs $\langle x, z_1 \cdots z_p\rangle$, where $x \in \Sigma^*$ and $z_1, \ldots, z_p \in \Sigma^r$, for which there exists a string $w \in \Sigma^p$ having even parity such that

$$\langle x, y_1, z_1\rangle \in A_{w_1}, \ldots, \langle x, y_p, z_p\rangle \in A_{w_p}. \tag{13}$$

- B_1 is the language of all pairs $\langle x, z_1 \cdots z_p\rangle$, where $x \in \Sigma^*$ and $z_1, \ldots, z_p \in \Sigma^r$, for which there exists a string $w \in \Sigma^p$ having odd parity such that

$$\langle x, y_1, z_1\rangle \in A_{w_1}, \ldots, \langle x, y_p, z_p\rangle \in A_{w_p}. \tag{14}$$

Given that A_0 and A_1 are disjoint and contained in $\mathcal{C}$, along with the fact that $\mathcal{C}$ is closed under joins and truth-table reductions, it follows that $B_0, B_1 \in \mathcal{C}$. The lemma now follows from the observation that

$$g(x) = \left|\left\{z \in \Sigma^s \,:\, \langle x, z\rangle \in B_0\right\}\right| - \left|\left\{z \in \Sigma^s \,:\, \langle x, z\rangle \in B_1\right\}\right| \tag{15}$$

for all $x \in \Sigma^*$, where $s = p \cdot r$. □

Lemma 6 *Let $\mathcal{C}$ be a nontrivial complexity class of languages that is closed under joins and polynomial-time truth table reductions, let $f_0, f_1 \in Gap \cdot \mathcal{C}$, and let p and q be polynomially bounded functions. For every string $x \in \Sigma^*$ and $y \in \Sigma_1^q$, define the matrix $M_{x,y}$ as*

$$\begin{aligned} \mathrm{Re}\big(\langle z|M_{x,y}|w\rangle\big) &= f_0(x, y, z, w), \\ \mathrm{Im}\big(\langle z|M_{x,y}|w\rangle\big) &= f_1(x, y, z, w), \end{aligned} \tag{16}$$

for all $z, w \in \Sigma^p$. In other words, each $M_{x,y}$ is a matrix whose entries are indexed by $z, w \in \Sigma^p$. There exist Gap $\cdot \mathcal{C}$ *functions g_0 and g_1 satisfying*

$$\begin{aligned} \mathrm{Re}\big(\langle z|M_{x,y_1} \cdots M_{x,y_q}|w\rangle\big) &= g_0(x, z, w), \\ \mathrm{Im}\big(\langle z|M_{x,y_1} \cdots M_{x,y_q}|w\rangle\big) &= g_1(x, z, w), \end{aligned} \tag{17}$$

for all $x \in \Sigma^*$ and $z, w \in \Sigma^p$, where $y_1, \ldots, y_q$ denote the elements of Σ_1^q sorted in lexicographic order.

Proof By the assumptions on $\mathcal{C}$ stated in the lemma, there must exist a Gap $\cdot \mathcal{C}$ function h satisfying

$$\begin{aligned} h(x, y, 0z, 0w) &= f_0(x, y, z, w), \\ h(x, y, 0z, 1w) &= f_1(x, y, z, w), \\ h(x, y, 1z, 0w) &= -f_1(x, y, z, w), \\ h(x, y, 1z, 1w) &= f_0(x, y, z, w), \end{aligned} \tag{18}$$

for all $x \in \Sigma^*$, $y \in \Sigma_1^q$, and $z, w \in \Sigma^p$. The matrix $N_{x,y}$ defined as

$$\langle u|N_{x,y}|v\rangle = h(x, y, u, v) \tag{19}$$

for all $x \in \Sigma^*$, $y \in \Sigma_1^q$ and $u, v \in \Sigma^{p+1}$ may be visualized as a 2×2 block matrix:

$$N_{x,y} = \begin{pmatrix} \mathrm{Re}(M_{x,y}) & \mathrm{Im}(M_{x,y}) \\ -\mathrm{Im}(M_{x,y}) & \mathrm{Re}(M_{x,y}) \end{pmatrix}. \tag{20}$$

We observe that

$$N_{x,y_1} \cdots N_{x,y_q} = \begin{pmatrix} \mathrm{Re}\big(M_{x,y_1} \cdots M_{x,y_q}\big) & \mathrm{Im}\big(M_{x,y_1} \cdots M_{x,y_q}\big) \\ -\mathrm{Im}\big(M_{x,y_1} \cdots M_{x,y_q}\big) & \mathrm{Re}\big(M_{x,y_1} \cdots M_{x,y_q}\big) \end{pmatrix}. \tag{21}$$

Given that h is a Gap $\cdot \mathcal{C}$ function, there must exist a Gap $\cdot \mathcal{C}$ function F for which

$$F(x, u_0 \cdots u_q, y_k) = h(x, y_k, u_{k-1}, u_k) \tag{22}$$

for all $x \in \Sigma^*$, $u_0, \ldots, u_q \in \Sigma^{p+1}$, and $k \in \{1, \ldots, q\}$.

Finally, define

$$G(x, u_0 \cdots u_q) = \prod_{y \in \Sigma_1^q} F(x, u_0 \cdots u_q, y) = h(x, y_1, u_0, u_1) \cdots h(x, y_q, u_{q-1}, u_q) \tag{23}$$

for all $x \in \Sigma^*$ and $u_0, \ldots, u_q \in \Sigma^{p+1}$, as well as

$$\begin{aligned} g_0(x, z, w) &= \sum_{u \in \Sigma^{(q-1)(p+1)}} G(x, 0zu0w), \\ g_1(x, z, w) &= \sum_{u \in \Sigma^{(q-1)(p+1)}} G(x, 0zu1w), \end{aligned} \tag{24}$$

for all $x \in \Sigma^*$ and $z, w \in \Sigma^p$. It follows by Lemmas 4 and 5 that $g_0, g_1 \in \mathrm{Gap} \cdot \mathcal{C}$.

Observing that g_0 and g_1 satisfy the (17), which is perhaps most evident from the (21), the proof of the lemma is complete. □

6.1 Quantum information and quantum circuits

The notation we use when discussing quantum information is standard for the subject, and we refer the reader to the books [30, 34, 38, 39] for further details. A couple of points concerning quantum information notation and conventions that may be helpful to some readers follow.

First, when we refer to a *register* X, we mean a collection of qubits that we wish to view as a single entity, and we then use the same letter $\mathcal{X}$ in a scripted font to denote the finite-dimensional complex Hilbert space associated with X (i.e., the space of complex vectors having entries indexed by binary strings of length equal to the number of qubits in X). The set of density operators acting on such a space is denoted $\mathrm{D}(\mathcal{X})$.

Second, a *channel* transforming a register X into a register Y is a completely positive and trace-preserving linear map $\mathcal{P}hi$ that transforms each density operator $\rho \in \mathrm{D}(\mathcal{X})$ into a density operator $\mathcal{P}hi(\rho) \in \mathrm{D}(\mathcal{Y})$. (More generally, such a mapping $\mathcal{P}hi$ transforms arbitrary linear operators acting on $\mathcal{X}$ into linear operators acting on $\mathcal{Y}$.) The *adjoint* of such a channel $\mathcal{P}hi$ is the uniquely determined linear map $\mathcal{P}hi^*$ transforming linear operators acting on $\mathcal{Y}$ into linear operators acting on $\mathcal{X}$ that satisfies the equation

$$\mathrm{Tr}\big(P\mathcal{P}hi(\rho)\big) = \mathrm{Tr}\big(\mathcal{P}hi^*(P)\rho\big) \tag{25}$$

for all density operators $\rho \in \mathrm{D}(\mathcal{X})$ and all positive semidefinite operators P acting on $\mathcal{Y}$. The adjoint $\mathcal{P}hi^*$ of a channel $\mathcal{P}hi$ is not necessarily itself a channel, but rather is a completely positive and *unital* linear map, which means that $\Phi^*(\mathbb{1}_{\mathcal{Y}}) = \mathbb{1}_{\mathcal{X}}$ (for $\mathbb{1}_{\mathcal{X}}$ and $\mathbb{1}_{\mathcal{Y}}$ denoting the identity operators acting on $\mathcal{X}$ and $\mathcal{Y}$, respectively). Intuitively speaking, if P is a measurement operator in the equation above, one can think of $\mathcal{P}hi^*$ as transforming the measurement operator P into a new measurement operator $\mathcal{P}hi^*(P)$, with the probability of this outcome for the state ρ being the same as if one first applied $\mathcal{P}hi$ to ρ and then measured with respect to P.

Now we will move on to *quantum circuits*, which are acyclic networks of quantum gates connected by qubit wires. We choose to use the standard, general model of quantum information based on density operators and quantum channels, as opposed to the restricted model of pure state vectors and unitary operations, when discussing quantum circuits. In this general model, each gate represents a quantum channel acting on a constant number of qubits—including nonunitary gates, such as gates that introduce fresh initialized qubits or gates that discard qubits. Through this model, ordinary classical circuits, as well as classical circuits that introduce randomness into

computations, can be viewed as special cases of quantum circuits. One may also represent measurements directly as quantum gates or circuits.

It is well-known that this general model is equivalent to the purely unitary model, as is explained in [1] and [37], for instance. The main benefits of using the general model in the context of this paper are that (i) it allows us to avoid having to constantly distinguish between input qubits and ancillary qubits, or output qubits and garbage qubits, and (ii) it has the minor but nevertheless positive side effect of eliminating the appearance of the irrational number $1/\sqrt{2}$ in many of the formulas that will appear.

We choose a universal gate set from which all quantum circuits are assumed to be composed. The gates in this set include *Hadamard*, *Toffoli*, and *phase-shift* gates (which induce the single-qubit unitary transformation determined by the actions $|0\rangle \mapsto |0\rangle$ and $|1\rangle \mapsto i|1\rangle$), as well as *ancillary gates* and *erasure gates*. Ancillary gates take no input qubits and output a single qubit in the $|0\rangle$ state, while erasure gates take one input qubit and produce no output qubits, and are described by the partial trace. Any other choice for the unitary gates that is universal for quantum computing could be taken instead, but the gate set just specified is both simple and convenient.

The *size* of a quantum circuit is defined to be the number of gates in the circuit plus the total number of input and output qubits. Thus, if a quantum circuit were to be represented in a standard way as a directed acyclic graph, its size would simply be the number of vertices, including a vertex for each input and output qubit, of the corresponding graph.

A collection $\{Q_x : x \in \Sigma^*\}$ of quantum circuits is said to be *polynomial-time generated* if there exists a polynomial-time deterministic Turing machine that, on input $x \in \Sigma^*$, outputs an encoding of the circuit Q_x. When such a family is parameterized by tuples of strings, it is to be understood that we are implicitly referring to one of the tuple-functions discussed previously. We will not have any need to discuss the specifics of the encoding scheme that we use, but naturally it is assumed to be efficient, with the size of a circuit and its encoding length being polynomially related.

The following lemma relates the complexity of computing circuit transition amplitudes to GapP functions. The essential idea it expresses is due to Fortnow and Rogers [21], who proved a variant of it for unitary computations by quantum Turing machines. An idea, similar in spirit, also appears in [15]. While a result along the lines of the lemma that follows is suggested in the survey paper [37], that paper does not include a proof, and so we include one below.

Lemma 7 *Let $\{Q_x : x \in \Sigma^*\}$ be a polynomial-time generated family of quantum circuits, where each circuit Q_x takes n input qubits and outputs k qubits, for polynomially bounded functions n and k. There exists a polynomially bounded function r and GapP functions f_0 and f_1 such that*

$$\begin{aligned} Re\big(\langle u|Q_x\big(|z\rangle\langle w|\big)|v\rangle\big) &= 2^{-r} f_0(x,z,w,u,v), \\ Im\big(\langle u|Q_x\big(|z\rangle\langle w|\big)|v\rangle\big) &= 2^{-r} f_1(x,z,w,u,v), \end{aligned} \tag{26}$$

for all $x \in \Sigma^*$, $z, w \in \Sigma^n$, and $u, v \in \Sigma^k$.

Proof Consider first an arbitrary channel $\mathcal{P}hi$ that maps n-qubit density operators to k-qubit density operators. The action of $\mathcal{P}hi$ on density operators is linear, and can therefore be represented through matrix multiplication. One concrete way to do this is to use the so-called natural representation (also known as the linear representation) of quantum channels.

A description of the natural representation of a quantum channel begins with the *vectorization* mapping: assuming M is a matrix whose rows and columns are indexed by strings of some length m, the corresponding vector $\mathrm{vec}(M)$ is indexed by strings of length $2m$ according to the following definition:

$$\mathrm{vec}(M) = \sum_{y,z \in \Sigma^m} \langle y|M|z\rangle \, |yz\rangle. \tag{27}$$

In words, the vectorization map reshapes a matrix into a vector by transposing the rows of the matrix into column vectors and stacking them on top of one another.

With respect to the vectorization mapping, the action of the channel $\mathcal{P}hi$ is described by its *natural representation* $K(\mathcal{P}hi)$, which is a linear mapping that acts as

$$K(\mathcal{P}hi)\mathrm{vec}(\rho) = \mathrm{vec}(\mathcal{P}hi(\rho)) \tag{28}$$

for every n-qubit density operator ρ. As a matrix, $K(\mathcal{P}hi)$ has columns indexed by strings of length $2n$ and rows indexed by strings of length $2k$. Its entries are described explicitly by the equation

$$\langle uv|K(\mathcal{P}hi)|zw\rangle = \langle u|\mathcal{P}hi(|z\rangle\langle w|)|v\rangle \tag{29}$$

holding for every $z, w \in \Sigma^n$ and $u, v \in \Sigma^k$. The (26) may therefore be equivalently written as

$$\begin{aligned} \mathrm{Re}\big(\langle uv|K(Q_x)|zw\rangle\big) &= 2^{-r} f_0(x, z, w, u, v), \\ \mathrm{Im}\big(\langle uv|K(Q_x)|zw\rangle\big) &= 2^{-r} f_1(x, z, w, u, v). \end{aligned} \tag{30}$$

It must be observed that the natural representation is multiplicative, in the sense that channel composition corresponds to matrix multiplication: $K(\mathcal{P}hi\mathcal{P}si) = K(\mathcal{P}hi)K(\mathcal{P}si)$ for all channels $\mathcal{P}hi$ and $\mathcal{P}si$ for which the composition $\mathcal{P}hi\mathcal{P}si$ makes sense. It is also helpful to note that a channel $\mathcal{P}hi(\rho) = U\rho U^*$ corresponding to a unitary operation has as its natural representation the operator

$$K(\mathcal{P}hi) = U \otimes \overline{U}. \tag{31}$$

Now let us turn to the family $\{Q_x : x \in \Sigma^*\}$. Because this family is polynomial-time generated, there must exist a polynomially bounded function r for which $\mathrm{size}(Q_x) \leq r$ for all $x \in \Sigma^*$. We may therefore write

$$Q_x = Q_{x,r} \cdots Q_{x,1} \tag{32}$$

for $Q_{x,1}, \ldots, Q_{x,r}$ being either identity channels or channels that describe the action of a single gate of Q_x tensored with the identity channel on all of the qubits besides the inputs of the corresponding gate that exist at the moment that the gate is applied. We also observe that the number of input qubits and output qubits of each $Q_{x,k}$ must be bounded by r.

Given that

$$K(Q_x) = K(Q_{x,r}) \cdots K(Q_{x,1}), \tag{33}$$

we are led to consider the natural representation of each channel $Q_{x,k}$. It will be convenient to identify each operator $K(Q_{x,k})$ with the matrix indexed by strings of length $2r$, as opposed to being indexed by strings whose lengths depend on the number of qubits in existence before and after $Q_{x,k}$ is applied, simply by padding $K(Q_{x,k})$ with rows and columns of zero entries.

The natural representations of the individual gates in the universal gate set we have selected are as follows:

1. Hadamard gate:

$$\frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \tag{34}$$

2. Phase gate:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -i & 0 & 0 \\ 0 & 0 & i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{35}$$

3. Toffoli gate:

$$\begin{pmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&1&0 \end{pmatrix} \otimes \begin{pmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&1&0 \end{pmatrix} \tag{36}$$

4. Ancillary qubit gate:

$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{37}$$

5. Erasure gate:

$$(1 \quad 0 \quad 0 \quad 1) \tag{38}$$

Based on these representations, it is straightforward to define GapP functions (or, in fact, FP functions) g_0 and g_1 such that

$$\begin{aligned} \mathrm{Re}\big(\langle uv|K(Q_{x,k})|zw\rangle\big) &= \frac{1}{2}\, g_0(x, z, w, u, v, y_k), \\ \mathrm{Im}\big(\langle uv|K(Q_{x,k})|zw\rangle\big) &= \frac{1}{2}\, g_1(x, z, w, u, v, y_k), \end{aligned} \tag{39}$$

for all $x \in \Sigma^*$, $k \in \{1, \ldots, r\}$, and $u, v, z, w \in \Sigma^r$, where we write $y_1, \ldots, y_r$ to denote the elements of Σ_1^r sorted in lexicographic order. It now follows through a

straightforward application of Lemma 6 there must exist GapP functions f_0 and f_1 satisfying (26) and therefore (30), for all $x \in \Sigma^*$, $z, w \in \Sigma^n$, and $u, v \in \Sigma^k$, as required. □

7.1 A tail bound for operator-valued random variables

We will make use of the following tail bound on the minimum eigenvalue of the average of a collection of operator-valued random variables. This bound follows from a more general result due to Tropp. In particular, the bound stated in the theorem below follows from Theorem 5.1 of [36] together with Pinsker's inequality, which relates the relative entropy of two distributions to their total variation distance.

Theorem 8 (Tropp) *Let d and N be positive integers, let $\eta \in [0, 1]$ and $\varepsilon > 0$ be real numbers, and let $X_1, \ldots, X_N$ be independent and identically distributed operator-valued random variables having the following properties:*

1. *Each X_k takes $d \times d$ positive semidefinite operator values satisfying $X_k \leq \mathbb{1}$.*
2. *The minimum eigenvalue of the expected operator $\mathrm{E}(X_k)$ satisfies $\lambda_{\min}(\mathrm{E}(X_k)) \geq \eta$.*

It is the case that

$$Pr\left(\lambda_{\min}\left(\frac{X_1 + \cdots + X_N}{N}\right) < \eta - \varepsilon\right) \leq d \exp(-2N\varepsilon^2). \tag{40}$$

9 Complexity classes for one-turn quantum refereed games

In this section we define the complexity classes to be considered in this paper: QRG(1), CQRG(1), and MQRG(1). The definitions of these classes all refer to the notion of a *referee*, which (in this paper) is a polynomial-time generated family

$$R = \{R_x : x \in \Sigma^*\} \tag{41}$$

of quantum circuits having the following special form

1. For each $x \in \Sigma^*$, the inputs to the circuit R_x are grouped into two registers: an n-qubit register A and an m-qubit register B, for polynomially bounded functions n and m.
2. The output of each circuit R_x is a single qubit, which is to be measured with respect to the standard basis immediately after the circuit is run.

Given that classical probabilistic states may be viewed as special cases of quantum states (corresponding to diagonal density operators), this definition of a referee can still be used in the situation in which either or both of the registers A and B is constrained to initially store a classical state.

We are interested in the situation that, for a given choice of an input string $x \in \Sigma^*$, the input to the circuit R_x is a product state of the form $\rho \otimes \sigma$, where $\rho \in \mathrm{D}(\mathcal{A})$ is a state of the register A and $\sigma \in \mathrm{D}(\mathcal{B})$ is a state of the register B. The state $\rho \in \mathrm{D}(\mathcal{A})$

is to be viewed as representing the state that Alice plays, while $\sigma \in \mathrm{D}(\mathcal{B})$ represents the state Bob plays. When the single output qubit of the circuit R_x is measured with respect to the standard basis, the outcome 1 is interpreted as "Alice wins," while the outcome 0 is interpreted as "Bob wins."

Now, consider the quantity defined as

$$\omega(R_x) = \max_{\rho \in \mathrm{D}(\mathcal{A})} \min_{\sigma \in \mathrm{D}(\mathcal{B})} \langle 1 | R_x (\rho \otimes \sigma) | 1 \rangle. \tag{42}$$

Given that $\mathrm{D}(\mathcal{A})$ and $\mathrm{D}(\mathcal{B})$ are compact and convex sets, and the value $\langle 1 | R_x (\rho \otimes \sigma) | 1 \rangle$ is bilinear in ρ and σ, Sion's min-max theorem implies that changing the order of the minimum and maximum does not change the value of the expression. That is, this quantity may alternatively be written

$$\omega(R_x) = \min_{\sigma \in \mathrm{D}(\mathcal{B})} \max_{\rho \in \mathrm{D}(\mathcal{A})} \langle 1 | R_x (\rho \otimes \sigma) | 1 \rangle. \tag{43}$$

This value represents the probability that Alice wins the game defined by the circuit R_x, assuming both Alice and Bob play optimally. With that definition in hand, we may now define the complexity class QRG(1), which is short for *one-turn quantum refereed games*.

Definition 9 A promise problem $A = (A_{\mathrm{yes}}, A_{\mathrm{no}})$ is contained in the complexity class $\mathrm{QRG}(1)_{\alpha,\beta}$ if there exists a referee $R = \{R_x : x \in \Sigma^*\}$ such that the following properties are satisfied:

1. For every string $x \in A_{\mathrm{yes}}$, it is the case that $\omega(R_x) \geq \alpha$.
2. For every string $x \in A_{\mathrm{no}}$, it is the case that $\omega(R_x) \leq \beta$.

We also define $\mathrm{QRG}(1) = \mathrm{QRG}(1)_{2/3,1/3}$.

In this definition, α and β may be constants, or they may be functions of the length of the input x. A short summary of known facts and observations concerning the complexity class QRG(1) follows.

- $\mathrm{QMA} \subseteq \mathrm{QRG}(1)$. This is because the referee's measurement may simply ignore Bob's state σ and treat Alice's state ρ as a quantum proof in a QMA proof system.
- $\mathrm{QRG}(1)$ is closed under complementation: $\mathrm{QRG}(1) = \text{co-}\mathrm{QRG}(1)$. For a promise problem $(A_{\mathrm{yes}}, A_{\mathrm{no}}) \in \mathrm{QRG}(1)$, one may obtain a one-turn quantum refereed game for $(A_{\mathrm{no}}, A_{\mathrm{yes}})$ by simply exchanging the roles of Alice and Bob.
- It is the case that $\mathrm{QRG}(1) = \mathrm{QRG}(1)_{\alpha,\beta}$ for a wide range of choices of α and β, similar to error bounds for BPP, BQP, and QMA. In particular, $\mathrm{QRG}(1) = \mathrm{QRG}(1)_{\alpha,\beta}$ provided that α and β are polynomial-time computable and satisfy

$$\alpha \leq 1 - 2^{-p}, \quad \beta \geq 2^{-p}, \quad \text{and} \quad \alpha - \beta \geq \frac{1}{p} \tag{44}$$

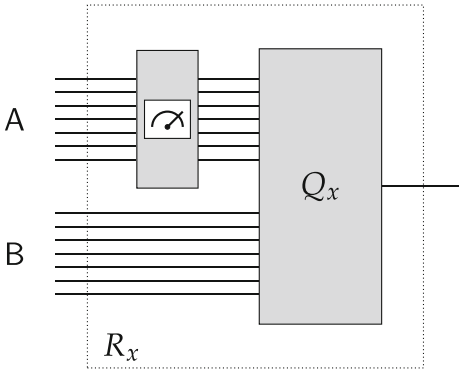


Fig. 1 A CQRG(1) referee. The register A is initially measured (or, equivalently, dephased) with respect to the standard basis, causing a classical state to be input into Q_x, along with the register B, which is unaffected by this standard basis measurement

for some choice of a strictly positive polynomially bounded function p.[2]

- QRG(1) $\subseteq$ PSPACE [28].

9.1 Definitions of new complexity classes

The first variant of QRG(1) we define is one in which Alice's state is restricted to be a classical state. We will call this class CQRG(1).

Definition 10 A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is contained in the complexity class $\text{CQRG}(1)_{\alpha,\beta}$ if there exists a referee $R = \{R_x : x \in \Sigma^*\}$ such that the following properties are satisfied:

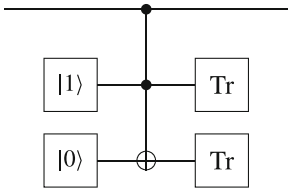
1. For every string $x \in \Sigma^*$, the circuit R_x takes the form illustrated in Fig. 1. That is, R_x takes an n-qubit register A and an m-qubit register B as input, measures each qubit of A with respect to the standard basis, leaving it in a classical state, and then runs the circuit Q_x on the pair (A, B), producing a single output qubit.
2. For every string $x \in A_{\text{yes}}$, it is the case that $\omega(R_x) \geq \alpha$.
3. For every string $x \in A_{\text{no}}$, it is the case that $\omega(R_x) \leq \beta$.

We also define $\text{CQRG}(1) = \text{CQRG}(1)_{2/3,1/3}$.

Formally speaking, the standard basis measurement suggested by Definition 10 can be implemented by independently performing the completely dephasing channel

[2]Error reduction may be performed through parallel repetition followed by majority vote. An analysis of this method for QRG(1) requires that one considers the possibility that the dishonest player (meaning the one that should not have a strategy that wins with high probability) entangles his or her state across the different repetitions, with the claimed bounds following from a similar analysis to parallel repetition followed by majority vote for QMA [30]. We note that there is no "in place" error reduction method known for QRG(1) that is analogous to the technique of [33] for QMA.

on each qubit of A. This channel can be constructed using the universal gate set we have selected using a Toffoli gate with suitably initialized inputs as follows:



Here the square labeled $|0\rangle$ is an ancillary gate, the square labeled $|1\rangle$ denotes an ancillary gate composed with a not-gate $X = HPPH$ (for H and P denoting Hadamard and phase-shift gates), and the square labeled Tr denotes an erasure gate.

In effect, a referee R that satisfies the first requirement of Definition 10 forces the state Alice plays to be a classical state (i.e., a state represented by a diagonal density operator). That is, for any density operator ρ that Alice might choose to play, the state of A that is input into Q_x takes the form

$$\sum_{y \in \Sigma^n} p(y) \, |y\rangle\langle y| \tag{45}$$

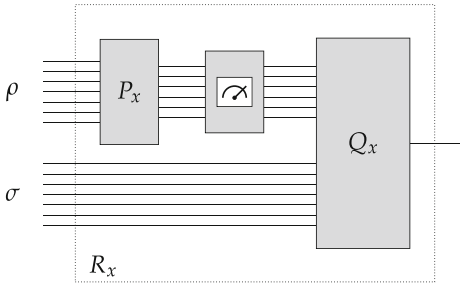
for some probability vector p over n-bit strings, and therefore the state that is plugged into the top n qubits of the circuit Q_x represents a classical state. Given that the standard basis measurement acts trivially on all diagonal states, we observe that Alice may cause an arbitrary diagonal density operator of the form (45) to be input into Q_x. In short, the set of possible states that may be input into the top n qubits of the circuit Q_x is precisely the set of diagonal n-qubit density operators.

The second variant of QRG(1) we define is one in which Alice and Bob both send quantum states to the referee, but the referee first measures Alice's state, obtaining a classical outcome, which is then measured together with Bob's state (as illustrated in Fig. 2).

Definition 11 A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is contained in the complexity class $\text{MQRG}(1)_{\alpha,\beta}$ if there exists a referee $R = \{R_x : x \in \Sigma^*\}$ such that the following properties are satisfied:

1. For every string $x \in \Sigma^*$, the circuit R_x takes the form illustrated in Fig. 2. That is, R_x takes an n-qubit register A and an m-qubit register B as input, and first applies

Fig. 2 An MQRG(1) referee



a quantum circuit P_x to A, yielding a k-qubit register Y, for k a polynomially bounded function. The register Y is then measured with respect to the standard basis, so that it then contains a classical state, and finally a quantum circuit Q_x is applied to the pair (Y, B), yielding a single qubit.

2. For every string $x \in A_{\text{yes}}$, it is the case that $\omega(R_x) \geq \alpha$.
3. For every string $x \in A_{\text{no}}$, it is the case that $\omega(R_x) \leq \beta$.

We also define $\text{MQRG}(1) = \text{MQRG}(1)_{2/3,1/3}$.

In essence, an MQRG(1) referee measures Alice's qubits with respect to a general, efficiently implementable measurement, which yields a k-bit classical outcome, which is then plugged into Q_x along with Bob's quantum state.

It is of course immediate that

$$\text{CQRG}(1) \subseteq \text{MQRG}(1) \subseteq \text{QRG}(1); \tag{46}$$

a CQRG(1) referee is a special case of an MQRG(1) referee in which P_x is the identity map on n qubits, while an MQRG(1) referee is a special case of a QRG(1) referee. We also observe that both CQRG(1) and MQRG(1) are robust with respect to error bounds in the same way as was described above for QRG(1).

10 Upper-bound on CQRG(1)

In this section, we prove that CQRG(1) is contained in $\exists \cdot \text{PP}$. The proof represents a fairly direct application of the Althöfer–Lipton–Young [2, 32] technique, although (as was suggested above) the quantum setting places a new demand on this technique that requires the use of a tail bound on sums of matrix-valued random variables. We will split the proof of this containment into two lemmas, followed by a short proof of the main theorem—this is done primarily because the lemmas will also be useful for proving $\text{MQRG}(1) \subseteq \text{P} \cdot \text{PP}$ in the section following this one. Some readers may wish to skip to the statement and proof of Theorem 14 below, as it explains the purpose of these two lemmas within the context of that theorem.

The first lemma represents an implication of Theorem 8 due to Tropp to the setting at hand.

Lemma 12 *Let k and m be positive integers, let $p \in \mathcal{P}(\Sigma^k)$ be a probability distribution on k-bit strings, let S_y be a $2^m \times 2^m$ positive semidefinite operator satisfying $0 \leq S_y \leq \mathbb{1}$ for each $y \in \Sigma^k$, and let $N \geq 72(m+2)$. For strings $y_1, \ldots, y_N \in \Sigma^k$ sampled independently from the distribution p, it is the case that*

$$Pr\left(\lambda_{\min}\left(\frac{S_{y_1} + \cdots + S_{y_N}}{N}\right) < \lambda_{\min}\left(\sum_{y \in \Sigma^k} p(y) S_y\right) - \frac{1}{12}\right) < \frac{1}{3}. \tag{47}$$

Proof Define $X_1, \ldots, X_N$ to be independent and identically distributed operator-valued random variables, each taking the (operator) value S_y with probability $p(y)$,

for every $y \in \Sigma^k$. The expected value of each of these random variables is therefore given by

$$P = \sum_{y \in \Sigma^k} p(y) S_y. \tag{48}$$

By taking $\eta = \lambda_{\min}(P)$ and $\varepsilon = 1/12$ in Theorem 8, we find that

$$\Pr\left(\lambda_{\min}\left(\frac{X_1 + \cdots + X_N}{N}\right) < \lambda_{\min}(P) - \frac{1}{12}\right) \leq 2^m \exp\left(-\frac{N}{72}\right) < \frac{1}{3}, \tag{49}$$

which is equivalent to the bound stated in the lemma. □

The second lemma uses counting complexity to relate the minimum eigenvalue of measurement operators defined by quantum circuits to PP languages. We note that the technique of weakly estimating the largest eigenvalue of a measurement operator using the trace of a power of that operator, through the relations (53) appearing in the proof below, is the essential idea behind the unpublished proof of the containment QMA ⊆ PP claimed in [31].

Lemma 13 *Let $\{Q_x : x \in \Sigma^*\}$ be a polynomial-time generated family of quantum circuits, where each circuit Q_x takes as input a k-qubit register Y and an m-qubit register B, for polynomially bounded functions k and m, and outputs a single qubit. For each $x \in \Sigma^*$ and $y \in \Sigma^k$, define an operator*

$$S_{x,y} = \big(\langle y| \otimes \mathbb{1}_{\mathcal{B}}\big) Q_x^* (|1\rangle\langle 1|) Q_x \big(|y\rangle \otimes \mathbb{1}_{\mathcal{B}}\big). \tag{50}$$

For every polynomially bounded function N, there exists a language $B \in \mathrm{PP}$ for which the following implications are true for all $x \in \Sigma^*$ and $y_1, \ldots, y_N \in \Sigma^k$:

$$\lambda_{\min}\left(\frac{S_{x,y_1} + \cdots + S_{x,y_N}}{N}\right) \geq \frac{2}{3} \quad \Rightarrow \quad (x, y_1 \cdots y_N) \in B, \tag{51}$$

$$\lambda_{\min}\left(\frac{S_{x,y_1} + \cdots + S_{x,y_N}}{N}\right) \leq \frac{1}{3} \quad \Rightarrow \quad (x, y_1 \cdots y_N) \notin B. \tag{52}$$

Proof The essence of the proof is that if P is an operator whose entries have real and imaginary parts proportional to GapP functions, and r is a polynomially bounded function, then there exists a GapP function that is proportional to the real part of $\mathrm{Tr}(P^r)$. When P is a $2^m \times 2^m$ positive semidefinite operator, this allows one to choose r to be sufficiently large, but still polynomially bounded, so that a GapP function is obtained that takes positive or negative values in accordance with the required implications (51) and (52), through the use of the following bounds relating the largest eigenvalue and the trace of any such P:

$$\lambda_{\max}(P)^r = \lambda_{\max}(P^r) \leq \mathrm{Tr}(P^r) \leq 2^m \lambda_{\max}(P^r) = 2^m \lambda_{\max}(P)^r. \tag{53}$$

In the case at hand, it will suffice to take $r = 2m$.

In greater detail, let us begin by defining

$$T_{x,y} = \big(\langle y| \otimes \mathbb{1}_{\mathcal{B}}\big) Q_x^* (|0\rangle\langle 0|) Q_x \big(|y\rangle \otimes \mathbb{1}_{\mathcal{B}}\big) \tag{54}$$

for each $x \in \Sigma^*$ and $y \in \Sigma^k$. Observe that $S_{x,y}$ and $T_{x,y}$ are positive semidefinite operators satisfying $S_{x,y} + T_{x,y} = \mathbb{1}_{\mathcal{B}}$, so that the implication in the statement of the lemma may alternatively be written as

$$\lambda_{\max}\left(\frac{T_{x,y_1} + \cdots + T_{x,y_N}}{N}\right) \leq \frac{1}{3} \quad \Rightarrow \quad (x, y_1 \cdots y_N) \in B, \tag{55}$$

$$\lambda_{\max}\left(\frac{T_{x,y_1} + \cdots + T_{x,y_N}}{N}\right) \geq \frac{2}{3} \quad \Rightarrow \quad (x, y_1 \cdots y_N) \notin B. \tag{56}$$

Thus, if the operator

$$P_{x,y_1 \cdots y_N} = \frac{T_{x,y_1} + \cdots + T_{x,y_N}}{N} \tag{57}$$

satisfies $\lambda_{\max}(P_{x,y_1 \cdots y_N}) \leq 1/3$, then

$$\mathrm{Tr}(P_{x,y_1 \cdots y_N}^{2m}) \leq \frac{2^m}{3^{2m}} < \frac{1}{3^m} \tag{58}$$

while if $\lambda_{\max}(P_{x,y_1 \cdots y_N}) \geq 2/3$, then

$$\mathrm{Tr}(P_{x,y_1 \cdots y_N}^{2m}) \geq \left(\frac{2}{3}\right)^{2m} > \frac{1}{3^m}. \tag{59}$$

By Lemma 7 there exists a polynomially bounded function r along with GapP functions f and g satisfying

$$\begin{aligned} \mathrm{Re}(\langle z|T_{x,y}|w\rangle) &= \mathrm{Re}\big(\langle 0|Q_x\big(|yz\rangle\langle yw|\big)|0\rangle\big) = 2^{-r} f(x, y, z, w), \\ \mathrm{Im}(\langle z|T_{x,y}|w\rangle) &= -\mathrm{Im}\big(\langle 0|Q_x\big(|yz\rangle\langle yw|\big)|0\rangle\big) = 2^{-r} g(x, y, z, w), \end{aligned} \tag{60}$$

for all $x \in \Sigma^*$, $y \in \Sigma^k$, and $z, w \in \Sigma^m$. Define functions F and G as follows:

$$\begin{aligned} F(x, y_1 \cdots y_N, z, w) &= f(x, y_1, z, w) + \cdots + f(x, y_N, z, w), \\ G(x, y_1 \cdots y_N, z, w) &= g(x, y_1, z, w) + \cdots + g(x, y_N, z, w), \end{aligned} \tag{61}$$

for all $x \in \Sigma^*$, $y_1, \ldots, y_N \in \Sigma^k$, and $z, w \in \Sigma^m$. It is the case that F and G are GapP functions satisfying

$$\begin{aligned} F(x, y_1 \cdots y_N, z, w) &= 2^r \cdot N \cdot \mathrm{Re}(\langle z|P_{x,y_1 \cdots y_N}|w\rangle), \\ G(x, y_1 \cdots y_N, z, w) &= 2^r \cdot N \cdot \mathrm{Im}(\langle z|P_{x,y_1 \cdots y_N}|w\rangle). \end{aligned} \tag{62}$$

Through an application of Lemmas 4 and 6, we conclude that there must exist a GapP function H satisfying

$$H(x, y_1 \cdots y_N) = 2^{2rm} \cdot N^{2m} \cdot \mathrm{Tr}\big(P_{x,y_1 \cdots y_N}^{2m}\big). \tag{63}$$

The GapP function

$$K(x, y_1 \cdots y_N) = 2^{2rm} \cdot N^{2m} - 3^m \cdot H(x, y_1 \cdots y_N) \tag{64}$$

therefore takes positive values if $\lambda_{\max}(P_{x,y_1 \cdots y_N}) \leq 1/3$, and takes negative values if $\lambda_{\max}(P_{x,y_1 \cdots y_N}) \geq 2/3$, implying the existence of a PP language B as claimed. □

Theorem 14 *$CQRG(1) \subseteq \exists \cdot PP$.*

Proof Let $A = (A_{\text{yes}}, A_{\text{no}})$ be any promise problem contained in CQRG(1), let a referee be fixed that establishes the inclusion $A \in \text{CQRG}(1)_{3/4,1/4}$, and let $\{Q_x : x \in \Sigma^*\}$ be the collection of circuits that describes this referee, in accordance with Definition 10.

Let $x \in A_{\text{yes}} \cup A_{\text{no}}$ be any input string. Consider first the situation that Alice plays deterministically, sending a string $y \in \Sigma^n$ to the referee, so that $\rho = |y\rangle\langle y|$. Having selected a state ρ representing Alice's play, we are effectively left with a binary-valued measurement being performed on the state sent to the referee by Bob. We observe that, for any choice of a state $\sigma \in \text{D}(\mathcal{B})$ representing Bob's play, the probabilities that the referee's measurement generates the outcomes 0 and 1 are given by

$$\langle 0|Q_x(|y\rangle\langle y| \otimes \sigma)|0\rangle \quad \text{and} \quad \langle 1|Q_x(|y\rangle\langle y| \otimes \sigma)|1\rangle, \tag{65}$$

respectively. By defining an operator $S_{x,y} \in \text{Pos}(\mathcal{B})$ as

$$S_{x,y} = \big(\langle y| \otimes \mathbb{1}_{\mathcal{B}}\big) Q_x^*(|1\rangle\langle 1|)\big(|y\rangle \otimes \mathbb{1}_{\mathcal{B}}\big), \tag{66}$$

we therefore obtain the measurement operator corresponding to the 1 outcome of this measurement, as

$$\text{Tr}\big(S_{x,y}\sigma\big) = \langle 1|Q_x(|y\rangle\langle y| \otimes \sigma)|1\rangle \tag{67}$$

and

$$\text{Tr}\big((\mathbb{1}_{\mathcal{B}} - S_{x,y})\sigma\big) = \langle 0|Q_x(|y\rangle\langle y| \otimes \sigma)|0\rangle \tag{68}$$

for all $\sigma \in \text{D}(\mathcal{B})$.

Now, as Bob aims to minimize the probability for outcome 1 to appear, the relevant property of the operator $S_{x,y}$ is its *minimum eigenvalue* $\lambda_{\min}(S_{x,y})$. A large minimum eigenvalue means that Alice has managed to force the outcome 1 to appear, regardless of what state Bob plays, whereas a small minimum eigenvalue means that Bob has at least one choice of a state that causes the outcome 1 to appear with small probability. Stated in more precise terms, Bob's optimal strategy in the case that Alice plays $\rho = |y\rangle\langle y|$ is to play any state $\sigma \in \text{D}(\mathcal{B})$ whose image is contained in the eigenspace of $S_{x,y}$ corresponding to the minimum eigenvalue $\lambda_{\min}(S_{x,y})$, which leads to a win for Alice with probability equal to this minimum eigenvalue and a win for Bob with probability $1 - \lambda_{\min}(S_{x,y})$.

In general, Alice will not play deterministically, but will instead play a distribution of strings $p \in \mathcal{P}(\Sigma^n)$. In this case, the resulting measurement operator on Bob's space becomes

$$\sum_{y \in \Sigma^n} p(y) S_{x,y}. \tag{69}$$

That is to say, the probability that Alice wins when she plays a distribution $p \in \mathcal{P}(\Sigma^n)$, and Bob plays optimally against this distribution, is given by the expression

$$\lambda_{\min}\left(\sum_{y \in \Sigma^n} p(y) S_{x,y}\right). \tag{70}$$

Determining whether x is a yes-instance or a no-instance of A is therefore equivalent to discriminating between the case that there exists a distribution $p \in \mathcal{P}(\Sigma^n)$

for which the minimum eigenvalue (70) is at least $3/4$ and the case in which this minimum eigenvalue is at most $1/4$ for all choices of $p \in \mathcal{P}(\Sigma^n)$.

The goal of the proof is to show that this decision problem is contained in $\exists \cdot \mathrm{PP}$. The $\exists$ operator will represent the existence or non-existence of a distribution $p \in \mathcal{P}(\Sigma^n)$ for which the minimum eigenvalue (70) is large, while a PP predicate will allow for an estimation of this minimum eigenvalue itself. A challenge that must be overcome in making this approach work is that using the $\exists$ operator in this way requires Alice's strategy to have a polynomial-length representation. However, given that a distribution $p \in \mathcal{P}(\Sigma^n)$ may have support that is exponentially large in n, an explicit description of p will generally have exponential size, assuming that the individual probabilities $p(y)$ are represented with a polynomial number of bits of precision.

This obstacle may be overcome using the Althöfer–Lipton–Young [2, 32] technique mentioned in the introduction: in place of a distribution $p \in \mathcal{P}(\Sigma^n)$, we consider an N-tuple of strings $(y_1, \ldots, y_N)$, representing N possible deterministic plays for Alice, for $N = N(|x|)$ being a suitable polynomially bounded function of the input length. This N-tuple will represent the distribution $q \in \mathcal{P}(\Sigma^n)$ obtained by selecting $j \in \{1, \ldots, N\}$ uniformly at random and then outputting the string y_j. That is, the distribution $q \in \mathcal{P}(\Sigma^n)$ represented by the N-tuple $(y_1, \ldots, y_N)$ is given by

$$q(y) = \frac{\big|\{j \in \{1, \ldots, N\} \,:\, y = y_j\}\big|}{N} \tag{71}$$

for each $y \in \Sigma^n$. Naturally, most choices of a distribution $p \in \mathcal{P}(\Sigma^n)$ are far away from any such distribution q. Nevertheless, the existence of a distribution $p \in \mathcal{P}(\Sigma^n)$ for which the minimum eigenvalue (70) is large does in fact imply the existence of an N-tuple $(y_1, \ldots, y_N)$ for which the distribution $q \in \mathcal{P}(\Sigma^n)$ defined by (71) is still a good play for Alice, meaning that the minimum eigenvalue

$$\lambda_{\min}\left(\frac{S_{x,y_1} + \cdots + S_{x,y_N}}{N}\right) \tag{72}$$

is also large, provided N is sufficiently large. This is precisely the content of Lemma 12.

In particular, by choosing $N = 72(m+2)$, where m is the number of qubits of B, we find that if the minimum eigenvalue (70) is at least 3/4, then with probability at least 2/3 (over the random choices of $y_1, \ldots, y_N$) the minimum eigenvalue (72) is at least 2/3. Of course, this implies the existence of an N-tuple $(y_1, \ldots, y_N)$ for which the minimum eigenvalue (72) is at least 2/3.

Naturally, if $x \in A_{\mathrm{no}}$, then the minimum eigenvalue (70) is at most $1/4$ for all choices of $p \in \mathcal{P}(\Sigma^n)$, and therefore it must be that

$$\lambda_{\min}\left(\frac{S_{x,y_1} + \cdots + S_{x,y_N}}{N}\right) \leq \frac{1}{4} < \frac{1}{3} \tag{73}$$

for all N-tuples $(y_1, \ldots, y_N)$. This is because the distribution q defined by (71) is simply one example of a distribution in $\mathcal{P}(\Sigma^n)$.

The purpose of Lemma 13 is now evident, for it states that there exists a language $B \in \mathrm{PP}$ such that if the minimum eigenvalue (72) is at least $2/3$, then $(x, y_1 \cdots y_N) \in$

B, while if this minimum eigenvalue is at most $1/3$, then $(x, y_1 \cdots y_N) \notin B$. Consequently, if $x \in A_{\text{yes}}$, then there exists a string $y_1 \cdots y_N \in \Sigma^{nN}$ such that $(x, y_1 \cdots y_N) \in B$, while if $x \in A_{\text{no}}$, then for every string $y_1 \cdots y_N \in \Sigma^{nN}$ it is the case that $(x, y_1 \cdots y_N) \notin B$. It has therefore been proved that $A \in \exists \cdot \text{PP}$ as required. □

11 Upper-bound on MQRG(1)

We now turn to the complexity class MQRG(1), and prove the containment $\text{MQRG}(1) \subseteq \text{P} \cdot \text{PP}$. In order to do this, we will first introduce a QMA-operator that, in some sense, functions in a way that is similar to the $\exists$ and P operators previously discussed.

Definition 15 For a given complexity class $\mathcal{C}$, the complexity class $\text{QMA} \cdot \mathcal{C}$ contains all promise problems $A = (A_{\text{yes}}, A_{\text{no}})$ for which there exists a polynomial-time generated family of quantum circuits $\{P_x : x \in \Sigma^*\}$, where each P_x takes $n = n(|x|)$ input qubits and outputs $k = k(|x|)$ qubits, along with a language $B \in \mathcal{C}$, such that the following implications hold.

1. If $x \in A_{\text{yes}}$, then there exists a density operator ρ on n qubits for which

$$\Pr\big(P_x(\rho) \in B\big) \geq \frac{2}{3}. \tag{74}$$

2. If $x \in A_{\text{no}}$, then for every density operator ρ on n qubits,

$$\Pr\big(P_x(\rho) \in B\big) \leq \frac{1}{3}. \tag{75}$$

Here, the notation $P_x(\rho) \in B$ refers to the event that P_x is applied to the state ρ, the output qubits are measured with respect to the standard basis, and the resulting string is contained in the language B. Figure 3 illustrates the associated process, with χ_B being the characteristic function of B on inputs of length k.

Theorem 16 *If $\mathcal{C}$ is nontrivial complexity class of languages that is closed under joins and truth-table reductions, then $\text{QMA} \cdot \mathcal{C} \subseteq \text{P} \cdot \mathcal{C}$.*

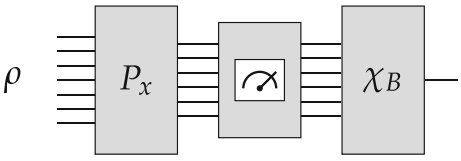


Fig. 3 Definition 15 is concerned with the probability that the output of a circuit P_x, measured with respect to the standard basis, is contained in the language B, assuming the input is ρ

Proof Let $A = (A_{\text{yes}}, A_{\text{no}}) \in \text{QMA} \cdot \mathcal{C}$, and let $\{P_x : x \in \Sigma^*\}$ be a polynomial-time generated family of quantum circuits that, together with a language $B \in \mathcal{C}$, establishes this inclusion according to Definition 15.

By Lemma 7 there exists a polynomially bounded function r and GapP functions f_0 and f_1 such that

$$\begin{aligned} \text{Re}\big(\langle u|P_x\big(|z\rangle\langle w|\big)|v\rangle\big) &= 2^{-r} f_0(x, z, w, u, v), \\ \text{Im}\big(\langle u|P_x\big(|z\rangle\langle w|\big)|v\rangle\big) &= 2^{-r} f_1(x, z, w, u, v), \end{aligned} \tag{76}$$

for all $x \in \Sigma^*$, $z, w \in \Sigma^n$, and $u, v \in \Sigma^k$. Define

$$\begin{aligned} g_0(x, z, w, u) &= \begin{cases} f_0(x, z, w, u, u) & \text{if } u \in B \\ 0 & \text{if } u \notin B, \end{cases} \\ g_1(x, z, w, u) &= \begin{cases} f_1(x, z, w, u, u) & \text{if } u \in B \\ 0 & \text{if } u \notin B, \end{cases} \end{aligned} \tag{77}$$

for all $x \in \Sigma^*$, $z, w \in \Sigma^n$, and $u \in \Sigma^k$. By the properties of $\mathcal{C}$, it is the case that $g_0, g_1 \in \text{Gap} \cdot \mathcal{C}$.

Next, define

$$\begin{aligned} F_0(x, z, w) &= \sum_{u \in \Sigma^k} g_0(x, z, w, u), \\ F_1(x, z, w) &= -\sum_{u \in \Sigma^k} g_1(x, z, w, u), \end{aligned} \tag{78}$$

for all $x \in \Sigma^*$ and $z, w \in \Sigma^n$. By Lemma 4 we have that $F_0, F_1 \in \text{Gap} \cdot \mathcal{C}$. We observe that

$$\begin{aligned} \text{Re}\big(\langle w|R_x|z\rangle\big) &= 2^{-r} F_0(x, z, w), \\ \text{Im}\big(\langle w|R_x|z\rangle\big) &= 2^{-r} F_1(x, z, w) \end{aligned} \tag{79}$$

for all $x \in \Sigma^*$ and $z, w \in \Sigma^n$, where

$$R_x = \sum_{u \in \Sigma^k \cap B} P_x^*\big(|u\rangle\langle u|\big). \tag{80}$$

Now let us consider the cases $x \in A_{\text{yes}}$ and $x \in A_{\text{no}}$. If $x \in A_{\text{yes}}$ then $\lambda_{\max}(R_x) \geq 2/3$, while if $x \in A_{\text{no}}$ then $\lambda_{\max}(R_x) \leq 1/3$. Observing that R_x is a positive semidefinite operator on a 2^n dimensional space, we have that

$$\lambda_{\max}(R_x)^{n+1} = \lambda_{\max}(R_x^{n+1}) \leq \text{Tr}(R_x^{n+1}) \leq 2^n \lambda_{\max}(R_x^{n+1}) = 2^n \lambda_{\max}(R_x)^{n+1}, \tag{81}$$

similar to (53) in the proof of Lemma 13. By Lemma 6 it follows that there exists a $\text{Gap} \cdot \mathcal{C}$ function G possessing the following properties.

1. If $x \in A_{\text{yes}}$ then

$$G(x) = 2^{(n+1)r} \text{tr}(R_x^{n+1}) \geq \frac{2^{(n+1)r+n+1}}{3^{n+1}} \tag{82}$$

2. If $x \in A_{\text{no}}$ then

$$G(x) = 2^{(n+1)r}\operatorname{tr}(R_x^{n+1}) \leq \frac{2^{(n+1)r+n}}{3^{n+1}}. \tag{83}$$

The Gap $\cdot\, \mathcal{C}$ function

$$H(x) = 3^{n+1}G(x) - 2^{(n+1)r+n} \tag{84}$$

therefore satisfies $H(x) > 0$ when $x \in A_{\text{yes}}$ and $H(x) \leq 0$ when $x \in A_{\text{no}}$. By Proposition 3 it follows that $A \in \mathrm{P} \cdot \mathcal{C}$. □

Next, we prove that MQRG(1) is contained in QMA·PP. Combining this fact with the previous theorem will establish the main result as an immediate corollary.

Theorem 17 *MQRG*(1) ⊆ *QMA* · *PP*.

Proof Consider any promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ in MQRG(1), and fix a referee that establishes the inclusion $A \in \mathrm{MQRG}(1)_{3/4,1/4}$. Let $\{P_x : x \in \Sigma^*\}$ and $\{Q_x : x \in \Sigma^*\}$ be a collection of circuits that describe this referee, in accordance with Definition 11. As in the proof of Theorem 14, define an operator

$$S_{x,y} = \big(\langle y| \otimes \mathbb{1}_{\mathcal{B}}\big) Q_x^*(|1\rangle\langle 1|)\big(|y\rangle \otimes \mathbb{1}_{\mathcal{B}}\big) \tag{85}$$

for each $x \in \Sigma^*$ and $y \in \Sigma^k$. If $x \in A_{\text{yes}}$, there must exists a state $\rho \in \mathrm{D}(\mathcal{A})$ such that

$$\lambda_{\min}\left(\sum_{y\in\Sigma^k} \langle y|P_x(\rho)|y\rangle S_{x,y}\right) \geq \frac{3}{4}, \tag{86}$$

while if $x \in A_{\text{no}}$, it is the case that

$$\lambda_{\min}\left(\sum_{y\in\Sigma^k} \langle y|P_x(\rho)|y\rangle S_{x,y}\right) \leq \frac{1}{4} \tag{87}$$

for every $\rho \in \mathrm{D}(\mathcal{A})$.

Now define a function $N = 72(m+2)$ and observe that N is polynomially bounded in $|x|$. By Lemma 13, there exists a language $B \in \mathrm{PP}$ for which these implications hold for all $x \in \Sigma^*$ and $y_1, \ldots, y_N \in \Sigma^k$:

$$\lambda_{\min}\left(\frac{S_{x,y_1} + \cdots + S_{x,y_N}}{N}\right) \geq \frac{2}{3} \quad \Rightarrow \quad (x, y_1 \cdots y_N) \in B, \tag{88}$$

$$\lambda_{\min}\left(\frac{S_{x,y_1} + \cdots + S_{x,y_N}}{N}\right) \leq \frac{1}{3} \quad \Rightarrow \quad (x, y_1 \cdots y_N) \notin B. \tag{89}$$

Finally, for each input x, define a circuit K_x that takes as input N registers $(\mathsf{A}_1, \ldots, \mathsf{A}_N)$, each consisting of n qubits, and outputs $N+1$ registers $(\mathsf{X}, \mathsf{Y}_1, \ldots, \mathsf{Y}_N)$. The register X is initialized to the state $|x\rangle\langle x|$, so that it simply echoes the input string x, and each register Y_j is obtained by independently applying the circuit P_x to A_j. Alternatively, one could write

$$K_x = |x\rangle\langle x| \otimes P_x^{\otimes N}, \tag{90}$$

with the understanding that we are identifying the state $|x\rangle\langle x|$ with the channel that inputs nothing and outputs the state $|x\rangle\langle x|$.

To prove that the promise problem A is contained in QMA · PP, it suffices to prove two things:

Completeness If it is the case that $x \in A_{\text{yes}}$, then there must exist a state $\xi \in \mathrm{D}(\mathcal{A}^{\otimes N})$ such that

$$\Pr(K_x(\xi) \in B) \geq \frac{2}{3}. \tag{91}$$

Soundness If it is the case that $x \in A_{\text{no}}$, then for every state $\xi \in \mathrm{D}(\mathcal{A}^{\otimes N})$ it must be that

$$\Pr(K_x(\xi) \in B) \leq \frac{1}{3}. \tag{92}$$

The proof of completeness follows a similar argument to the proof of Theorem 14. Let $\rho \in \mathrm{D}(\mathcal{A})$ be any state for which (86) is satisfied, and let $\xi = \rho^{\otimes N}$. It is evident that the output of $K_x(\xi)$ is given by $(x, y_1 \cdots y_N)$, for $y_1, \ldots, y_N \in \Sigma^k$ sampled independently from the distribution

$$p(y) = \langle y|P_x(\rho)|y\rangle. \tag{93}$$

It follows by Lemma 12 that

$$\Pr(K_x(\xi) \in B) \geq \frac{2}{3}. \tag{94}$$

For the proof of soundness, the possibility that the state $\xi \in \mathrm{D}(\mathcal{A}^{\otimes N})$ does not take product form must be considered. Our aim is to prove that if $y_1, \ldots, y_N$ are randomly selected according to the distribution that assigns the probability

$$\big\langle y_1 \cdots y_N \big| P_x^{\otimes N}(\xi) \big| y_1 \cdots y_N \big\rangle \tag{95}$$

to each tuple $(y_1, \ldots, y_N)$, then

$$\Pr\left(\lambda_{\min}\left(\frac{S_{x,y_1} + \cdots + S_{x,y_N}}{N}\right) \leq \frac{1}{3}\right) \geq \frac{2}{3}, \tag{96}$$

for this implies that $\Pr(K_x(\xi) \in B) \leq 1/3$ by (89). Toward this goal, choose a density operator $\sigma \in \mathrm{D}(\mathcal{B})$ for which

$$\sum_{y \in \Sigma^k} \langle y|P_x(\rho)|y\rangle \mathrm{Tr}\big(S_{x,y}\sigma\big) \leq \frac{1}{4} \tag{97}$$

for all $\rho \in \mathrm{D}(\mathcal{A})$, which is possible by Sion's min-max theorem under the assumption (87), and define random variables $Z_1, \ldots, Z_N$ as

$$Z_j = \mathrm{Tr}\big(S_{x,y_j}\sigma\big) \tag{98}$$

for every $j \in \{1, \ldots, N\}$, assuming that $y_1, \ldots, y_N$ are chosen at random as above. It suffices to prove that

$$\Pr\left(\frac{Z_1 + \cdots + Z_N}{N} \leq \frac{1}{3}\right) \geq \frac{2}{3}, \tag{99}$$

as we have $\lambda_{\min}(H) \leq \mathrm{Tr}(H\sigma)$ for all Hermitian operators H.

The complication we face at this point is that the random variables $Z_1, \ldots, Z_N$ are not necessarily independent (because ξ does not necessarily have product form), so the most standard form of Hoeffding's inequality will not suffice to establish the required bound (99). However, we observe that $Z_1, \ldots, Z_N$ are discrete random variables that take values in the interval [0, 1] and satisfy the inequality

$$\mathrm{E}(Z_j | Z_1 = \alpha_1, \ldots, Z_{j-1} = \alpha_{j-1}) \leq \frac{1}{4} \tag{100}$$

for all $j \in \{2, \ldots, N\}$ and $\alpha_1, \ldots, \alpha_{j-1} \in [0, 1]$ for which $\Pr(Z_1 = \alpha_1, \ldots, Z_{j-1} = \alpha_{j-1})$ is nonzero. This is evident from the inequality (97), for it must hold when ρ is equal to the reduced state of register A_j, conditioned on any choice of $y_1, \ldots, y_{j-1}$ (and therefore on any choice of values $Z_1 = \alpha_1, \ldots, Z_{j-1} = \alpha_{j-1}$) that appear with nonzero probability. While the standard statement of Hoeffding's inequality does not suffice for our needs, the standard *proof* of Hoeffding's inequality does establish that

$$\Pr\left(\frac{Z_1 + \cdots + Z_N}{N} \geq \frac{1}{3}\right) = \Pr\left(\frac{Z_1 + \cdots + Z_N}{N} \geq \frac{1}{4} + \frac{1}{12}\right) \leq \exp\left(-\frac{2N}{144}\right) < \frac{1}{3}, \tag{101}$$

as explained in an appendix at the end of the paper. Having obtained this bound, the proof is complete. □

Corollary 18 *MQRG*(1) $\subseteq$ P $\cdot$ *PP*.

12 Conclusion

We have proved containments on two restricted versions of QRG(1), which we have called CQRG(1) and MQRG(1) (Fig. 4). A diagram illustrating the containments is provided.

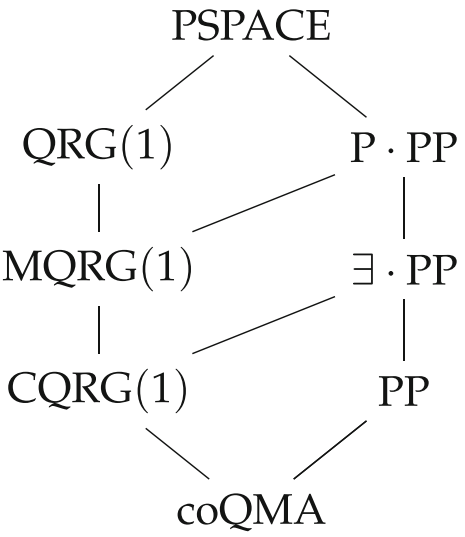


Fig. 4 Hasse diagram showing the relationships between CQRG(1), MQRG(1), and other complexity classes

The question that originally motivated the work reported in this paper is whether the containment QRG(1) $\subseteq$ PSPACE can be improved. We did not succeed in this endeavor, and so we leave this as an open question. Observing that the containments we prove establish that CQRG(1) and MQRG(1) are contained in the counting hierarchy, we ask specifically: is QRG(1) also contained in the counting hierarchy?

Appendix A: Hoeffding's inequality for dependent random variables with bounded conditional expectation

In the proof of Theorem 17 we used a slight variant of Hoeffding's inequality, where the assumption of independence is replaced by a bound on conditional expectation. We expect that a bound along these lines has been observed before, but we have not found a suitable reference. (A similar bound is proved in [4] for Bernoulli random variables, but we require the bound to hold more generally for discrete random variables.)

It is, however, straightforward to adapt the most typical proof of Hoeffding's inequality to obtain this bound, as we now explain. We begin with Hoeffding's lemma, which is the essential ingredient in the proof, and which we state without proof. (A proof may be found in [7], among many other references).

Lemma 19 *Let X be a random variable taking values in $[\alpha, \beta]$, for real numbers $\alpha < \beta$, and assume $\mathrm{E}(X) \leq 0$. For every $\lambda > 0$ it is the case that*

$$E\big(\exp(\lambda X)\big) \leq \exp\left(\frac{\lambda^2}{8(\beta - \alpha)^2}\right). \tag{102}$$

Remark 20 The more typical assumption for this lemma is that $\mathrm{E}(X) = 0$, but (as is not surprising) it is true assuming instead that $\mathrm{E}(X) \leq 0$. This follows immediately from the observation that if $\mathrm{E}(X) \leq 0$, then

$$\mathrm{E}(\exp(\lambda X)) \leq \mathrm{E}(\exp(\lambda(X - \mathrm{E}(X)))). \tag{103}$$

The next lemma provides the inequality in the proof of Hoeffding's inequality that would ordinarily follow from the assumption of independence. For simplicity we prove this lemma for discrete random variables, which suffices for our needs.

Lemma 21 *Let X and Y be discrete random variables taking values in $[\alpha, \beta]$ for real numbers $\alpha < \beta$, and assume that $E(Y \mid X) \leq 0$. For every $\lambda > 0$ it is the case that*

$$E\left(\exp(\lambda(X + Y)\right) \leq \exp\left(\frac{\lambda^2}{8(\beta - \alpha)^2}\right) E(\exp(\lambda X)). \tag{104}$$

Proof We may write

$$\mathrm{E}\left(\exp(\lambda(X+Y))\right) = \sum_x \exp(\lambda x)\mathrm{E}(\exp(\lambda Y) \mid X = x)\Pr(X = x), \tag{105}$$

where the sum ranges over all possible values of X. By the assumption $\mathrm{E}(Y \mid X) \leq 0$, Hoeffding's lemma implies

$$\begin{aligned}
&\sum_x \exp(\lambda x)\mathrm{E}(\exp(\lambda Y) \mid X = x)\Pr(X = x) \\
&\leq \exp\left(\frac{\lambda^2}{8(\beta-\alpha)^2}\right)\sum_x \exp(\lambda x)\Pr(X = x) \\
&= \exp\left(\frac{\lambda^2}{8(\beta-\alpha)^2}\right)\mathrm{E}(\exp(\lambda X)),
\end{aligned} \tag{106}$$

as required. □

Finally, we state and prove the variant of Hoeffding's inequality we have used (again for discrete random variables).

Theorem 22 *Let $X_1, \ldots, X_n$ be discrete random variables taking values in $[0, 1]$, let $\gamma \in [0, 1]$, and assume that*

$$E(X_k \mid X_1, \ldots, X_{k-1}) \leq \gamma \tag{107}$$

for all $k \in \{1, \ldots, n\}$. For all $\varepsilon > 0$ it is the case that

$$Pr\big(X_1 + \cdots + X_n \geq (\gamma + \varepsilon)n\big) \leq \exp(-2n\varepsilon^2). \tag{108}$$

Proof For every $\lambda > 0$ we have that

$$\begin{aligned}
&\Pr\big(X_1 + \cdots + X_n \geq (\gamma+\varepsilon)n\big) \\
&= \Pr\big(\exp\big(\lambda(X_1 + \cdots + X_n - \gamma n)\big) \geq \exp(\lambda\varepsilon n)\big) \\
&\leq \frac{\mathrm{E}\big(\exp\big(\lambda(X_1 + \cdots + X_n - \gamma n)\big)\big)}{\exp(\lambda\varepsilon n)}
\end{aligned} \tag{109}$$

by Markov's inequality. Applying Lemma 21 iteratively yields

$$\mathrm{E}\big(\exp\big(\lambda(X_1 + \cdots + X_n - \gamma n)\big)\big) \leq \exp\left(\frac{n\lambda^2}{8}\right). \tag{110}$$

Choosing $\lambda = 4\varepsilon$ yields the claimed bound. □

References

1. Aharonov, D., Kitaev, A., Nisan, N.: Quantum circuits with mixed states. In: Proceedings of the 30th Annual ACM Symposium on Theory of Computing, pp. 20–30 (1998)
2. Althöfer, I.: On sparse approximations to randomized strategies and convex combinations. Linear Algebra Appl. **199**, 339–355 (1994)
3. Babai, L.: Trading group theory for randomness. In: Proceedings of the 17th Annual ACM Symposium on Theory of Computing, pp. 421–429 (1985)

4. Babai, L., Cooperman, G., Finkelstein, L., Luks, E., Seress, Á.: Fast Monte Carlo algorithms for permutation groups. J. Comput. Syst. Sci. **50**, 296–307 (1995)
5. Babai, L., Moran, S.: Arthur-Merlin games: a randomized proof system, and a hierarchy of complexity classes. J. Comput. Syst. Sci. **36**(2), 254–276 (1988)
6. Beigel, R., Reingold, N., Spielman, D.: PP Is closed under intersection. J. Comput. Syst. Sci. **50**(2), 191–202 (1995)
7. Bhattacharya, R., Waymire, E.: A Basic Course in Probability Theory Springer second edition (2016)
8. Cai, J.-Y.: $S_2^P \subseteq ZPP^{NP}$. J. Comput. Syst. Sci. **73**(1), 25–35 (2007)
9. Canetti, R.: More on BPP and the polynomial-time hierarchy. Inf. Process. Lett. **57**(5), 237–241 (1996)
10. Condon, A., Feigenbaum, J., Lund, C., Shor, P.: Probabilistically checkable debate systems and approximation algorithms for PSPACE-hard functions. Chic. J. Theor. Comput. Sci. **1995**, 4 (1995)
11. Condon, A., Feigenbaum, J., Lund, C., Shor, P.: Random debaters and the hardness of approximating stochastic functions. SIAM J. Comput. **26**(2), 369–400 (1997)
12. Chandra, A., Kozen, D., Stockmeyer, L.: Alternation. J. ACM **28**(1), 114–133 (1981)
13. Condon, A.: Computational Models of Games. PhD thesis University of Washington (1987)
14. Demaine, E., Hearn, R.: Playing games with algorithms: Algorithmic combinatorial game theory. In: Albert, M., Nowakowski, R. (eds.) Games of No Chance 3, pp. 3–56. Cambridge University Press (2009)
15. Dawson, C.M., Hines, A.P., Mortimer, D., Haselgrove, H.L., Nielsen, M.A., Osborne, T.: Quantum Computing and Polynomial Equations over the Finite Field $\mathbb{Z}_2$. Quantum Inf. Comput. :102–112 (2005)
16. Fortnow, L., Impagliazzo, R., Kabanets, V., Umans, C.: On the complexity of succinct zero-sum games. Comput. Complex. **17**, 353–376 (2008)
17. Feige, U., Kilian, J.: Making games short. In: Proceedings of the 29th Annual ACM Symposium on Theory of Computing, pp. 506–516 (1997)
18. Feigenbaum, J., Koller, D., Shor, P.: A game-theoretic classification of interactive complexity classes. In: Proceedings of the 10th Conference on Structure in Complexity Theory, pp. 227–237 (1995)
19. Fortnow, L.: Counting complexity. In: Hemaspaandra, L., Selman, A. (eds.) Complexity Theory Retrospective II, pp. 81–107. Springer (1997)
20. Fortnow, L., Reingold, N.: PP Is closed under trhuth-table reductions. Inf. Comput. **124**(1), 1–6 (1996)
21. Fortnow, L., Rogers, J.: Complexity limitations on quantum computation. J. Comput. Syst. Sci. **59**(2), 240–252 (1999)
22. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. In: Proceedings of the 17th Annual ACM Symposium on Theory of Computing, pp. 291–304 (1985)
23. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. **18**(1), 186–208 (1989)
24. Gutoski, G.: Upper bounds for quantum interactive proofs with competing provers. In: Proceedings of the 20th Annual IEEE Conference on Computational Complexity, pp. 334–343 (2005)
25. Gutoski, G., Watrous, J.: Quantum interactive proofs with competing provers. In: Proceedings of the 22nd Symposium on Theoretical Aspects of Computer Science, volume 3404 of Lecture Notes in Computer Science, pp. 605–616. Springer (2005)
26. Gutoski, G., Watrous, J.: Toward a general theory of quantum games. In: Proceedings of the 39th Annual ACM Symposium on Theory of Computing, pp. 565–574 (2007)
27. Gutoski, G., Wu, X.: Parallel approximation of min-max problems. Comput. Complex. **2**(22), 385–428 (2013)
28. Jain, R., Watrous, J.: Parallel approximation of non-interactive zero-sum quantum games. In: Proceedings of the 24th IEEE Conference on Computational Complexity, pp. 243–253 (2009)
29. Koller, D., Megiddo, N.: The complexity of two-person zero-sum games in extensive form. Games Econ. Behav. **4**, 528–552 (1992)
30. Kitaev, A., Shen, A., Vyalyi, M.: Classical and Quantum Computation, volume 47 of Graduate Studies in Mathematics American Mathematical Society (2002)
31. Kitaev, A., Watrous, J.: Parallelization, amplification, and exponential time simulation of quantum interactive proof systems. In: Proceedings of the 32nd Annual ACM Symposium on Theory of Computing, pp. 608–617 (2000)
32. Lipton, R., Young, N.: Simple strategies for large zero-sum games with applications to complexity theory. In: Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, pp. 734–740 (1994)

33. Marriott, C., Watrous, J.: Quantum Arthur-Merlin games. Comput. Complex. **14**(2), 122–152 (2005)
34. Nielsen, M., Chuang, I.: Quantum computation and quantum information cambridge university press (2000)
35. Russell, A., Sundaram, R.: Symmetric alternation captures BPP. Comput. Complex. **7**(2), 152–162 (1998)
36. Tropp, J.: User-friendly tail bounds for sums of random matrices. Found. Comput. Math. **12**(4), 389–434 (2012)
37. Watrous, J.: Quantum Computational Complexity. In: Encyclopedia of Complexity and System Science. Springer (2009)
38. Watrous, J.: Theory of quantum information cambridge university press (2018)
39. Wilde, M.: Quantum Information Theory. Cambridge University Press second edition (2017)