Electronic Colloquium on Computational Complexity, Report No. 131 (2019)

A Simple Proof of Vyalyi's Theorem and some Generalizations

Lieuwe Vinkhuijzen
Universiteit Leiden
Email: l.t.vinkhuijzen@liacs.leidenuniv.nl

André Deutz
Universiteit Leiden
Email: a.h.deutz@liacs.leidenuniv.nl

***Abstract*—In quantum computational complexity theory, the class QMA models the set of problems efficiently verifiable by a quantum computer the same way that NP models this for classical computation. Vyalyi proved that if QMA = PP then PH ⊆ QMA. In this note, we give a simple, self-contained proof of the theorem, using only the closure properties of the complexity classes in the theorem statement. We then extend the theorem in two directions: (i) we strengthen the consequent, proving that if QMA = PP then QMA = $\mathbf{PH}^{\mathbf{PP}}$, and (ii) we weaken the hypothesis, proving that if QMA = co-QMA then PH ⊆ QMA. Lastly, we show that all the above results hold, without loss of generality, for the class QAM instead of QMA. We also formulate a "Quantum Toda's Conjecture".**

1. Introduction

A major open question in quantum computational complexity theory is to find the relationships between quantum complexity classes and classical ones. In particular, we do not know how **QMA** relates to the polynomial hierarchy or to **PP**. For the first problem, no containment is known in either direction. While it is known that **QMA** is contained in **PP**, it is open whether the inclusion is strict. Progress in this direction was made in 2003 when Vyalyi showed that the two questions are in fact related:

Theorem 1 (Vyalyi [1]). *If* $\mathbf{QMA} = \mathbf{PP}$ *then* $\mathbf{PH} \subseteq \mathbf{PP}$.

For the proof, see page 6. Vyalyi took this as evidence that $\mathbf{QMA} \neq \mathbf{PP}$ because otherwise $\mathbf{PH} \subseteq \mathbf{QMA}$, which seems unlikely.

Vyalyi's proof of Theorem 1 introduces a new complexity class called $\mathbf{A_0PP}$ and uses Gap functions to show that $\mathbf{QMA} \subseteq \mathbf{A_0PP} \subseteq \mathbf{PP}$; then it uses Gap functions and a strong version of Toda's Theorem to show that if $\mathbf{A_0PP} = \mathbf{PP}$ then $\mathbf{PH} \subseteq \mathbf{PP}$. Specifically, it uses the version of Toda's Theorem which states that $\mathbf{PH} \subseteq \mathbf{P}^{\#\mathbf{P}[1]}$: all languages in the polynomial hierarchy can be solved with only one query to a counting oracle.

Our new proof is, in our view, simpler. It has three ingredients: (i) the usual version of Toda's Theorem [4] (namely $\mathbf{PH} \subseteq \mathbf{P}^{\mathbf{PP}}$), (ii) that **PP** is closed under complement [3] and (iii) that $\mathbf{QMA} \cap$ **co-QMA** is closed under Turing reductions. The third ingredient is, to the best of our knowledge, new.

We anticipate the objection that our proof still uses Toda's Theorem and that therefore the complexity of the original proof is not eliminated, but merely outsourced. Our response is to give a *second*, wholly self-contained elementary proof, whose ideas we immediately use to improve Theorem 1 in two ways. First, we strengthen the consequent, as follows:

Theorem 2. *If* $\mathbf{QMA} = \mathbf{PP}$ *then* $\mathbf{QMA} = \mathbf{PH}^{\mathbf{PP}}$.

That is, the hypothesis implies that the polynomial hierarchy collapses relative to a counting oracle. Second, we weaken the hypothesis of Vyalyi's Theorem from $\mathbf{QMA} = \mathbf{PP}$ to merely $\mathbf{QMA} = \textbf{co-QMA}$:

Theorem 3. *If* $\mathbf{QMA} = \textbf{co-QMA}$ *then* $\mathbf{PH} \subseteq \mathbf{PP}$.

ISSN 1433-8092

2. Preliminaries

We assume that the reader is familiar with the basics of Computational Complexity, in particular the polynomial hierarchy (see Arora and Barak [5]) and with quantum computing (see Nielsen and Chuang [6] or Kitaev, Shin and Vyalyi [7]). We work with languages over the binary alphabet $\{0,1\}$.

A *Turing reduction* from a language L to a language K is an algorithm with oracle access to K which solves L. If L *Turing-reduces* to K, we write $L \leq_T K$. If the reduction algorithm runs in polynomial time, we say that there is a polynomial-time deterministic Turing reduction from L to K and write $L \leq_T^p K$ or $L \in \mathbf{P}^K$. A class $\mathfrak{C}$ is *closed under polynomial-time Turing reductions*, written $\mathfrak{C} = \mathbf{P}^{\mathfrak{C}}$, if $L \leq_T^p K$ implies $L \in \mathfrak{C}$ for every language $K \in \mathfrak{C}$ and $L \subseteq \{0,1\}^*$. A class is *closed under complement*, i.e. $\mathfrak{C} = \textbf{CO}-\mathfrak{C}$, if $L \in \mathfrak{C} \iff \overline{L} \in \mathfrak{C}$ for all $L \in \mathfrak{C}$, where $\overline{L} = \{0,1\}^* \setminus L$.

The class **QMA** was defined by Kitaev et al. [7] (they called it **BQNP**):

Definition 1 (The class **QMA**: Quantum Merlin-Arthur games)**.** *A language $L \subseteq \{0,1\}^*$ is in* **QMA** *if there are polynomials $m(n), w(n)$ and a polynomial-time constructible family of quantum circuits $\{U_x\}_{x\in\{0,1\}^*}$ receiving an $m(n)$-qubit input and using $w(n)$ qubits of workspace, possessing completeness and soundness:*

- ***Completeness:*** *If $x \in L$ then the circuit U_x accepts some input state $|\psi\rangle_{m(n)}$ with probability at least $1 - 2^{-n}$*
- ***Soundness:*** *If $x \notin L$ then the circuit U_x rejects all input states with probability at least $1 - 2^{-n}$*

The circuit family is called a protocol *for L.*

The class **QMA** is often studied as a set of *promise problems* (which are pairs (L_{yes}, L_{no}), where the algorithm is allowed to behave arbitrarily on inputs outside of $L_{yes} \cup L_{no}$), because in that context it allows for complete problems, notably the Local Hamiltonian Problem [7]. For us it will be more natural to consider **QMA** simply as a set of languages, because in this context the operation of using a language as an oracle is cleaner, but we stress that this decision is without loss of generality.

The class **PP**, or Probabilistic Polynomial time, was defined by Gill [3], who showed that **PP** is closed under complement. Toda's Theorem states that $\mathbf{PH} \subseteq \mathbf{P}^{\mathbf{PP}}$ [4].

3. Closure properties of QMA

In this section, we show that the class $\mathbf{QMA} \cap \textbf{CO-QMA}$ is closed under polynomial-time Turing reductions. That is,

Theorem 4. $\mathbf{QMA} \cap \textbf{CO-QMA} = \mathbf{P}^{\mathbf{QMA} \cap \textbf{CO-QMA}}$

For the proof, see page 4. The ideas in the proof are best illustrated by recalling two other theorems: (i) the class $\mathbf{NP} \cap \textbf{CO-NP}$ is closed under Turing reductions (Theorem 5), and (ii) the class **QMA** is closed under intersection (Theorem 7). To streamline the proof of Theorem 7, we introduce the Entanglement Independence Lemma, (Lemma 6).

The result $\mathbf{NP} \cap \textbf{CO-NP} = \mathbf{P}^{\mathbf{NP} \cap \textbf{CO-NP}}$ is classic, and the idea of the Entanglement Independence Lemma is simply the technique Kitaev et al. used for error amplification when they defined **QMA** [7].

Theorem 5. $\mathbf{NP} \cap \textbf{CONP}$ *is closed under polynomial-time deterministic Turing reductions:* $\mathbf{P}^{\mathbf{NP} \cap \textbf{CONP}} = \mathbf{NP} \cap \textbf{CONP}$.

Proof. The trivial direction is $\mathbf{NP} \cap \textbf{CO-NP} \subseteq \mathbf{P}^{\mathbf{NP} \cap \textbf{CO-NP}}$, so we will only show the other direction, $\mathbf{P}^{\mathbf{NP} \cap \textbf{CO-NP}} \subseteq \mathbf{NP} \cap \textbf{CO-NP}$. To this end, it is sufficient to show that $\mathbf{P}^{\mathbf{NP} \cap \textbf{CO-NP}} \subseteq \mathbf{NP}$ because **P** is closed under complement.

Let $K \in \mathbf{P}^{\mathbf{NP} \cap \textbf{CO-NP}}$ be a language decided by the polynomial-time (say $\mathcal{O}(t(n))$-time) Turing Machine M^L, with access to an oracle language $L \in \mathbf{NP} \cap \textbf{CO-NP}$. Because $L \in \mathbf{NP} \cap \textbf{CO-NP}$, there are non-deterministic Turing Machines Y and N recognizing the languages L and $\overline{L}$, respectively, both running in time $\mathcal{O}(t'(n))$. If we manage to simulate M^L with a non-deterministic machine, we will have proved the theorem.

Clearly *if* we manage to obtain the answers to all the queries M^L makes, *then* we can faithfully simulate M^L. The central insight is that we can obtain the answers by guessing and then verifying them. Therefore the algorithm will be as follows.

Before we compute anything, (i) we nondeterministically guess that the queries that M^L is going to make are the strings $s_1, \ldots, s_{t(n)}$, (ii) we guess the answers $a_1, \ldots, a_{t(n)}$ to all the queries that M^L makes, and lastly (iii) we guess certificate strings $y_1, \ldots, y_{t(n)}$ and $z_1, \ldots, z_{t(n)}$ for the machines Y and N, respectively.[1] The remainder of the computation is deterministic. The algorithm checks that its guesses were correct: for each i, it checks that $Y(s_i, y_i) = a_i$ and $N(s_i, z_i) = \neg a_i$ (meaning that, for example, if $a_2 = 1$ then Y accepts the certificate y_2 we guessed for s_2 and N rejects z_2). If any of these checks fail, we have evidently guessed incorrectly, and we reject on this computation path.

Lastly, we simulate M^L. When it makes the i-th query, we check that it queries $s_i \overset{?}{\in} L$; if so, we feed it the answer a_i and continue the simulation, but if not, we immediately reject, because we have incorrectly guessed which strings M^L would query. When M^L halts and accepts, we accept; otherwise we reject.

If M^L accepts, then our machine accepts. The accepting paths are exactly those that correctly guessed which strings M^L was going to query, what the answers were, and what certificates would satisfy Y and N. If M^L rejects, then our machine rejects too, because even the paths that obtained correct answers for the oracle queries still reject when M^L halts and rejects. □

The next lemma, the Entanglement Independence Lemma, says that if two **QMA** protocols possess soundness individually, then soundness is preserved when they are combined in a single circuit and their inputs are entangled, as long as they are implemented and measured independently, as in Figure 1. This is not obvious a priori, because oftentimes quantum circuits behave in surprising ways when clever use is made of entanglement. The result of the Lemma, then, is that in this case, no such clever use is possible for Merlin: entangling the two certificates for the two circuits gives him no advantage.

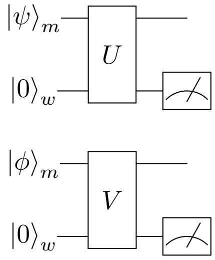

Figure 1. The circuits U and V receive inputs that may be entangled.

Lemma 6 (Entanglement Independence Lemma)**.** *Let U, V be two quantum circuits as in Figure 1, both receiving an m-qubit input and a w-qubit workspace, with measurement operators Π_U, Π_V. Suppose that U and V accept with probability at most a and b, respectively, regardless of their m-qubit input. Then the probability that both U and V accept when they are implemented jointly, and when their inputs may be entangled, is at most $a \cdot b$.*

Proof. The proof uses a trick by Marriot and Watrous [8]. They are able to express the probability that a circuit accepts an m-qubit input in terms of the eigenvalues of a $2^m \times 2^m$ matrix, as in Equation 1. Equation 1 follows from the equality $(I^{\otimes m} \otimes |0\rangle) \cdot |\psi\rangle_m = |\psi\rangle_m \otimes |0\rangle$.

$$\begin{aligned} & P[U \text{ accepts } |\psi\rangle] && (1)\\ &= \langle\psi| \langle 0| U^\dagger \Pi_U U |\psi\rangle |0\rangle \\ &= \langle\psi| \cdot \left((I^{\otimes m} \otimes \langle 0|) \cdot U^\dagger \Pi_U U \cdot (I^{\otimes m} \otimes |0\rangle) \right) |\psi\rangle \end{aligned}$$

Let

$$\begin{aligned} \tilde{U} &= (I^{\otimes m} \otimes \langle 0|) \cdot U^\dagger \Pi_U U \cdot (I^{\otimes m} \otimes |0\rangle) \\ \tilde{V} &= (I^{\otimes m} \otimes \langle 0|) \cdot V^\dagger \Pi_V V \cdot (I^{\otimes m} \otimes |0\rangle) \end{aligned}$$

Clearly these operators are Hermitian, so their eigenvalues are real, so their "largest" eigenvalue is well-defined.

Using Equation 1, we can express the probability that U and V both accept an $m{+}m$-qubit quantum state $|\phi\rangle = \sum_i y_i |a_i\rangle |b_i\rangle$ as $\langle\phi| (\tilde{U} \otimes \tilde{V}) |\phi\rangle$. This probability is maximized at the largest eigenvalue of $\tilde{U} \otimes \tilde{V}$. But the eigenvalues of $\tilde{U} \otimes \tilde{V}$ are exactly the products of the eigenvalues of $\tilde{U}$ and $\tilde{V}$. More precisely, for every pair of eigenvalues λ, μ of U and V, $\lambda \cdot \mu$ is an eigenvalue of $\tilde{U} \otimes \tilde{V}$.

[1] All these strings are polynomially-bounded: The strings s_i are not longer than $t(n)$ bits, because M^L runs in time $t(n)$ bits. The certificates y_i, z_i are not longer than $t'(t(n))$ bits.

By assumption, the largest eigenvalues of U and V are at most a and b, so the largest eigenvalue of $\tilde{U} \otimes \tilde{V}$ is at most $a \cdot b$. □

The approach of Marriot and Watrous has the advantage that the qubits of the workspace are encapsulated by the operator $\tilde{U}$, which allows us to express acceptance of U, and in turn express perfect play by Merlin, in terms of the eigenvalues of $\tilde{U}$:

$$\begin{aligned} &P[U \text{ accepts} \mid \text{Perfect play by Merlin}] \\ &= \max_{|\psi\rangle} \langle\psi| \tilde{U} |\psi\rangle \end{aligned}$$

This quantity is maximized at an eigenvalue of $\tilde{U}$, and the message that Merlin will send to maximize Arthur's probability of acceptance is a corresponding eigenvector. The significance is that one of the maximizing eigenvectors is an untangled state. This gives us exactly what we wanted: while Merlin has the ability to entangle his certificates, he gains no advantage from doing so compared to sending unentangled certificates.

Next, we use the Entanglement Independence Lemma to show that **QMA** is closed under intersection.

Theorem 7. *If $L, K \in$* **QMA** *then $L \cap K \in$* **QMA**. *That is,* **QMA** *is closed under intersection.*

Proof. Let $L \in$ **QMA** and $K \in$ **QMA**. For a particular input string $x \in \{0,1\}^*$, we now design a simple **QMA** protocol for membership in $L \cap K$. Let U and V be amplified circuits of the **QMA** protocols for the languages L and K, respectively. We ask Merlin for two certificates $|\psi_L\rangle_p$ and $|\psi_K\rangle_q$, feed them to the circuits and measure the outcomes, like in Figure 2. We accept iff both circuits accept. The difficulty is that Merlin can entangle the two certificates, but we will show that he gains no advantage from doing so.

Part 1: Completeness. Suppose that $x \in L \cap K$. Then Merlin can be honest and send us the state $|\psi_L\rangle_p \otimes |\psi_K\rangle_q$. These are two unentangled certificates for membership of L and K, so the two subcircuits U and V can be analyzed independently: U will accept with probability $|\Pi_U U |\psi_L\rangle|^2 \geq \ ^9/_{10}$ and V will accept with probability $|\Pi_V V |\psi_K\rangle|^2 \geq \ ^9/_{10}$. Therefore both circuits accept with probability at least $\left(\ ^9/_{10}\right)^2 \geq \ ^2/_3$, which suffices to show that the protocol has completeness.

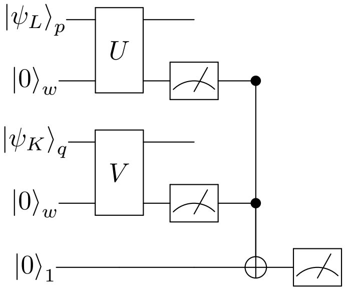

Figure 2. A circuit which receives two certificates $|\psi_L\rangle_p$ and $|\psi_K\rangle_q$, possibly entangled, and executes the protocols for languages L and K on them.

Part 2: Soundness. Suppose that $x \notin L \cap K$, e.g. because $x \notin L$. Then Merlin may send us any arbitrary, possibly highly entangled state. If U is implemented alone, then it accepts any certificate with probability at most $\ ^1/_3$. Here, we provide U with a well-initialised workspace of $|0\rangle_w$, so by the Entanglement Independence Lemma, U will also accept with low probability, at most $\ ^1/_3$, regardless of how Merlin entangles its input register with the rest of the certificate. Therefore the probability that the circuit as a whole accepts is also $\leq \ ^1/_3$, so the protocol has soundness. □

By a similar argument, **QMA** is closed under union. Lastly, we need the union bound in the following form.

Lemma 8 (Union Bound)**.** *If $t(n) \in \Omega(1)$ is a polynomial then*

$$\lim_{n \to \infty} (1 - 2^{-n})^{t(n)} = 1$$

□

We are now ready to prove Theorem 4.

Proof of Theorem 4. It suffices to give a **QMA** protocol for a language in $\mathbf{P}^{\mathbf{QMA} \cap \mathbf{co\text{-}QMA}}$, because **P** is closed under complement. So let M^L be a polynomial-time (say $t(n)$-time) Turing Machine with oracle access to a language $L \in$ **QMA** $\cap$ **co-QMA** and let $\{Y_s\}_{s \in \{0,1\}^*}$ and $\{N_s\}_{s \in \{0,1\}^*}$ be the circuit families corresponding to the **QMA** protocols for L and $\overline{L}$, respectively.

In the proof of Theorem 5, we fed M^L truthful answers to all its oracle queries. Here we will settle for something which suffices for our purposes: either (i) if $x \in L$, then with very high probability we will give M^L truthful answers to its queries, or else (ii) if $x \notin L$, with very high probability we will detect any attempt of Merlin to fool us, and reject. To this end, we assume that the circuits Y_s and N_s are amplified such that if $s \in L$, then there is a state that Y_s accepts with probability $\geq 1 - 2^{-n}$, whereas if $s \notin L$, then Y_s rejects every state with probability $\geq 1 - 2^{-n}$.

In this protocol, we expect that Merlin's message contains (i) the (classical) strings $s_1, \ldots, s_{t(n)}$ that M^L queries, (ii) their answers $a_1, \ldots, a_{t(n)}$ and (iii) quantum states $|\psi_1\rangle, \ldots, |\psi_{t(n)}\rangle$ and $|\phi_1\rangle, \ldots, |\phi_{t(n)}\rangle$ which will serve as the certificates to Y and N. This is the same strategy as pursued in the proof of Theorem 5, except now the certificates are quantum states. The certificates are the only parts of Merlin's message that need to be quantum, because parts (i) and (ii) are simply classical bit strings.

Before we simulate M^L, we measure all the qubits which we expect to be classical bits (the answers and the query strings) in the computational basis. Then we compute, for each i, whether Y_{s_i} accepts $|\psi_i\rangle$ and whether N_{s_i} accepts $|\phi_i\rangle$. We reject if Y_{s_i} does not output a_i or N_{s_i} does not output $\neg a_i$, just as before. Lastly, we simulate M^L just as in Theorem 5, rejecting if M^L queries a string we did not anticipate or if M^L rejects. Otherwise, we accept.

The only part where we use quantum computing is in the evaluation of the certificates, and we have to argue that this does not influence the soundness of the protocol.

Part 1: Completeness. Suppose that M^L accepts. Then Merlin can be honest and simply send us good, unentangled certificates for the algorithms Y_s and N_s. In particular, for any query s, if $s \in L$, then Y_s will accept the good certificate $|\psi_i\rangle$ with probability at least $1 - 2^{-n}$ and N_s will reject its certificate $|\phi_i\rangle$ with at least the same probability. Because these circuits are implemented independently of one another and the state $|\psi_i\rangle$ is not entangled with $|\phi_i\rangle$, these two events are independent, so the probability that both happen is at least $(1 - 2^{-n})^2$. This needs to happen for all queries, of which there are at most $t(n)$. The probability that all oracle queries proceed this way is at least

$$(1 - 2^{-n})^{2t(n)} \tag{2}$$

Because $t(n)$ is a polynomial, this quantity tends to 1 for large n, by Lemma 8, meaning that with probability tending to 1, we obtain correct answers for all queries. After that, simulating M^L is a deterministic computation and we copy its answer, so we output the correct answer with probability tending to 1.

Part 2: Soundness. Suppose that M^L rejects. Then Merlin will send us a possibly very complicated entangled state. We start by measuring all the registers which we expect to be classical in the computational basis, so that these registers are now truly classical bits and are not entangled with the rest of the certificate. The certificates to the queries remain quantum. The fact that the various certificates may be entangled with one another presents the principal hurdle in this proof, which we overcome by the Entanglement Independence Lemma.

Suppose that in our simulation we feed M^L only correct answers. Then we will certainly reject. So in order to make us accept, Merlin must make us compute the wrong answer for at least one query, which happens when $s \in L$ but Y_s rejects and N_s accepts. But according to the Entanglement Independence Lemma, and because the protocol N_s possesses soundness, the probability that N_s accepts any certificate state, no matter how the state is entangled with other parts of the circuit that N_s does not touch, is at most 2^{-n}. (Our assymetric focus here on the protocol N_s instead of Y_s is because it is easy enough for Merlin to make Y_s reject by sending a bad certificate. In the classical case he might send a non-satisfying assignment to the Satisfiability problem, for example).

Above we have described the "good event" that the circuit N_s rejects a single bad certificate. For our simulation to succeed, each query must be a "good event", that is, it must happen $t(n)$ times. By Lemma 8, this probability tends to 1. □

The following corollary is immediate.

Theorem 9. $\mathbf{QMA}$ *is closed under complement if and only if it is closed under Turing reductions. In symbols:*

$$\mathbf{QMA} = \mathbf{co\text{-}QMA} \iff \mathbf{QMA} = \mathbf{P}^{\mathbf{QMA}} \quad (3)$$

Proof. Suppose that $\mathbf{QMA} = \mathbf{co\text{-}QMA}$. Then $\mathbf{QMA} = \mathbf{QMA} \cap \mathbf{co\text{-}QMA} = \mathbf{P}^{\mathbf{QMA}\cap\mathbf{co\text{-}QMA}} = \mathbf{P}^{\mathbf{QMA}}$ by Theorem 4. In the other direction, if $\mathbf{QMA} = \mathbf{P}^{\mathbf{QMA}}$, then it is closed under complement, because $\mathbf{P}$ is closed under complement. □

4. A simple proof

Theorem 1 (Vyalyi [1])**.** *If* $\mathbf{QMA} = \mathbf{PP}$ *then* $\mathbf{PH} \subseteq \mathbf{PP}$.

Proof. Suppose that $\mathbf{QMA} = \mathbf{PP}$. Since $\mathbf{PP}$ is closed under complement [3], now $\mathbf{QMA}$ is also closed under complement, so by Theorem 9 it is closed under Turing reductions. By assumption, then, $\mathbf{PP}$ is closed under Turing reductions, i.e. $\mathbf{PP} = \mathbf{P}^{\mathbf{PP}}$. Toda's Theorem completes the proof: $\mathbf{P}^{\mathbf{PP}} \supseteq \mathbf{PH}$. □

5. Stronger statements

In this section, we improve Vyalyi's Theorem in two directions. First we strengthen the consequent from $\mathbf{PH} \subseteq \mathbf{PP}$ to $\mathbf{PP} = \mathbf{PH}^{\mathbf{PP}}$ (Theorem 2), and second, we weaken the hypothesis from $\mathbf{QMA} = \mathbf{PP}$ to merely $\mathbf{QMA} = \mathbf{co\text{-}QMA}$ (Theorem 3). Both proofs build on one more ingredient:

Theorem 10. $\mathbf{NP}^{\mathbf{QMA}\cap\mathbf{co\text{-}QMA}} \subseteq \mathbf{QMA}$.

Proof. We ask Merlin for (i) a certificate for the $\mathbf{NP}$ machine that we are simulating and (ii) for the queries and their answers similar to how we asked them in Theorem 4. The first thing the $\mathbf{QMA}$ machine does is measure the certificate in the computational basis. Merlin is supposed to send a classical string, so if he is honest, nothing happens. If he is dishonest, then the certificate collapses to some classical string, and the $\mathbf{NP}$ machine will reject this if it receives correct answers to its queries.

After measuring the certificate Merlin gave for the $\mathbf{NP}$ machine, the remaining computation is a simulation of a deterministic Turing Machine which makes queries to $\mathbf{QMA} \cap \mathbf{co\text{-}QMA}$, that is, it is a $\mathbf{P}^{\mathbf{QMA}\cap\mathbf{co\text{-}QMA}}$ computation. By Theorem 4, this can be simulated in $\mathbf{QMA}$. □

Theorem 11. *If* $\mathbf{QMA} = \mathbf{co\text{-}QMA}$ *then* $\mathbf{QMA} = \mathbf{PH}^{\mathbf{QMA}}$.

Proof. The non-trivial direction is $\mathbf{PH}^{\mathbf{QMA}} \subseteq \mathbf{QMA}$. The proof is by induction. For the base case, we prove that $\Sigma_1^{\mathbf{P}^{\mathbf{QMA}}} = \mathbf{QMA}$, using Theorem 10:

$$\begin{aligned} \Sigma_1^{\mathbf{P}^{\mathbf{QMA}}} &= \mathbf{NP}^{\mathbf{QMA}} \\ &= \mathbf{NP}^{\mathbf{QMA}\cap\mathbf{co\text{-}QMA}} \subseteq \mathbf{QMA} \end{aligned}$$

The last inclusion is Theorem 10. The induction step assumes that $\Sigma_i^{\mathbf{P}^{\mathbf{QMA}}} = \mathbf{QMA}$ and derives $\Sigma_{i+1}^{\mathbf{P}}{}^{\mathbf{QMA}} = \mathbf{QMA}$:

$$\begin{array}{rll} \Sigma_i^{\mathbf{P}^{\mathbf{QMA}}} & = \mathbf{QMA} & \text{Induction hypothesis} \\ \Sigma_{i+1}^{\mathbf{P}}{}^{\mathbf{QMA}} & = \mathbf{NP}^{\Sigma_i^{\mathbf{P}^{\mathbf{QMA}}}} & \text{By definition} \\ & = \mathbf{NP}^{\mathbf{QMA}} & \text{Induction hypothesis} \\ & = \mathbf{NP}^{\mathbf{QMA}\cap\mathbf{co\text{-}QMA}} & \mathbf{QMA} = \mathbf{co\text{-}QMA} \\ & & \text{by assumption} \\ & \subseteq \mathbf{QMA} & \text{Theorem 10} \end{array}$$

We have shown that $\mathbf{PH}^{\mathbf{QMA}} \subseteq \mathbf{QMA}$, i.e. relative to a $\mathbf{QMA}$ oracle, every level of the polynomial hierarchy is contained in the unrelativized version of $\mathbf{QMA}$. For the opposite inclusion, we note that $\mathbf{QMA} \subseteq \mathbf{P}^{\mathbf{QMA}} \subseteq \mathbf{PH}^{\mathbf{QMA}}$ is unconditional. □

All goals set out in the introduction are immediate corollaries of Theorem 11, using the facts that $\mathbf{PP}$ is closed under complement [3] and that $\mathbf{QMA} \subseteq \mathbf{PP}$ (see [1]). [2]

Theorem 2. *If* $\mathbf{QMA} = \mathbf{PP}$ *then* $\mathbf{QMA} = \mathbf{PH}^{\mathbf{PP}}$.

Proof. If $\mathbf{QMA} = \mathbf{PP}$, then $\mathbf{QMA} = \mathbf{co\text{-}QMA}$ since $\mathbf{PP}$ is closed under complement [3]. So by Theorem 11, $\mathbf{QMA} = \mathbf{PH}^{\mathbf{QMA}} = \mathbf{PH}^{\mathbf{PP}}$. □

Theorem 3. *If* $\mathbf{QMA} = \mathbf{co\text{-}QMA}$ *then* $\mathbf{PH} \subseteq \mathbf{PP}$.

[2] We cite [1] for the result $\mathbf{QMA} \subseteq \mathbf{PP}$, because it is the earliest published proof known to the authors. However, it is mentioned earlier, in the text that defines $\mathbf{QMA}$ [7], where it is left as an excercise to the reader.

Proof. If $\mathbf{QMA} = \textbf{co-QMA}$ then by Theorem 11, $\mathbf{PH} \subseteq \mathbf{PH}^{\mathbf{QMA}} = \mathbf{QMA}$. The Theorem follows from the inclusion $\mathbf{QMA} \subseteq \mathbf{PP}$ [1]. □

Note that we have fulfilled our promise of giving a self-contained proof of Vyalyi's Theorem that does not depend on Toda's Theorem, as each of Theorem 2 and 3 implies Vyalyi's Theorem.

We feel that these proofs go some way towards illustrating *why* Vyalyi's Theorem is true, namely: for the same reason as in the classical case, if we rephrase the classical case as follows.

Theorem 12. *If* $\mathbf{NP} = \textbf{co-NP}$ *then* $\mathbf{PH} \subseteq \mathbf{NP}$.

The idea, of course, is that if the unsatisfiability of a SAT formula always has a short certificate, then that enables one to check Σ_2^P predicates and, by induction, the whole polynomial hierarchy. In our quantum case, quantum unsatisfiability has a short quantum certificate, but the idea is the same.

6. Generalization to AM and QAM

The techniques developed to give the simple proof of Vyalyi's Theorem also apply to the classes **AM** and **QAM**. In particular, we reprove all the theorems with **QAM** in lieu of **QMA**. The following definition of **QAM** is due to Marriot and Watrous [8]. The class captures languages solvable by a two-round public-coin interactive proof in which Arthur sends a classical message to Merlin, who responds with a quantum state, which Arthur may use in a quantum computation.

Definition 2 (The class **QAM**). *A language $L \subseteq \{0,1\}^*$ is in* **QAM** *if there are polynomials $m(n), s(n)$ and a polynomial-time uniform family $\{U_x\}_{x\in\{0,1\}^*}$ of quantum circuits acting on three collections on qubits: Arthur's workspace qubits, $m(n)$ qubits sent by Merlin and $s(n)$ classical bits, corresponding to a sequence of coin-flips sent by Arthur to Merlin, on which Merlin's message may depend. The family must satisfy the following completeness and soundness conditions for all $x \in \{0,1\}^*$:*

Completeness: *If $x \in L$ then there is a collection of quantum states $\{|\psi_y\rangle\}_{y\in\{0,1\}^s}$ such that*

$$\frac{1}{2^{s(n)}}\sum_{y\in\{0,1\}^{s(n)}} \Pr\left[U_x \textit{ accepts } |\psi_y\rangle\,|y\rangle\right] \geq 1-2^{-n}$$

Soundness: *If $x \notin L$ then for every collection of quanum states $\{|\psi_y\rangle\}_{y\in\{0,1\}^*}$,*

$$\frac{1}{2^{s(n)}}\sum_{y\in\{0,1\}^{s(n)}} \Pr\left[U_x \textit{ accepts } |\psi_y\rangle\,|y\rangle\right] \leq 2^{-n}$$

Marriot and Watrous showed that the completeness and soundness parameter 2^{-n} may be replaced by a constant or by $2^{-r(n)}$ for a polynomial $r(n)$, without loss of generality.

In this Section, we follow the same setup as above. We will establish that $\mathbf{AM} \cap \textbf{co-AM}$ and $\mathbf{QAM} \cap \textbf{co-QAM}$ are closed under Turing reductions, and then that $\mathbf{NP}^{\mathbf{QAM}\cap\textbf{co-QAM}} \subseteq \mathbf{QAM}$ (Theorem 16). Using these results, the two new versions of Vyalyi's Theorem will be easy to obtain. All Theorems are analogues of their **QMA** counterpart, and the proofs are set up so as to deviate minimally from the corresponding proof above. For the sake of exposition, the theorems are first proven in the classical setting, for the class **AM**, after which the theorem is reproven in the quantum setting.

Theorem 13. $\mathbf{P}^{\mathbf{AM}\cap\textbf{co-AM}} = \mathbf{AM} \cap \textbf{co-AM}$.

Proof. Let K a language decided by deterministic $\mathcal{O}(t(n))$-time oracle Turing Machine with oracle access to a language $L \in \mathbf{AM} \cap \mathbf{coAM}$. If we find an **AM** protocol to simulate M^L, we will have proved the theorem.

Since $L \in \mathbf{AM} \cap \mathbf{coAM}$, there are deterministic Turing Machines Y and N which execute the **AM** protocols for L and $\overline{L}$, respectively, using $r(n)$ random bits, running in time $t(n)$ and erring with probability $\leq 2^{-n}$.

Again, of course, *if* we obtain the answers to all the queries M^L makes, *then* we can simulate M^L. We obtain those answers by running the machines Y and N, just as in the proof of Theorem 5, with two differences: (i) the protocol starts with Arthur generating some appropriate number of random bits and communicating those to Merlin, and (ii) the protocol may err with some small probability conditioned on those random bits.

The protocols starts with Arthur flipping $2t(n)\cdot r(t(n))$ coins and sending the result to Merlin. This number, $2t(n)\cdot r(t(n))$, is an upper bound on the number of random coins all the upcoming protocols need, as M^L makes at most $t(n)$ queries, each at most $t(n)$ bits long. Then Y and N will use $r(t(n))$ random bits each to answer a query of length $t(n)$. We expect Merlin to send us (i) the strings $q_1, \dots, q_{t(n)}$ that M^L will query given the random coins we just guessed, (ii) the answers $a_1, \dots, a_{t(n)}$ to those queries and (iii) (classical) certificates $y_1, \dots, y_{t(n)}$ and $z_1, \dots, z_{t(n)}$ for the machines Y and N, respectively.

The first step for Arthur is to check whether the certificates are good. For each i, he checks that $Y(q_i, s_i, y_i) = a_i$ and $N(q_i, s_i, z_i) = \neg a_i$. Here s_i is the string of random coins Arthur flipped at the beginning to feed to the i-th query. If any of these checks fail, he rejects.

If all checks pass, then Arthur simulates M^L as before: If the i-th query of M^L is not the string q_i, he rejects; otherwise, he feeds M^L the answer a_i and resumes the simulation. When M^L halts, he copies its answer as his output.

Part 1: Completeness. Suppose that M^L accepts. If good certificates to our queries exist (a good certificate is one that Y will accept if the answer is *yes*), then Merlin will send them and Arthur will feed M^L correct answers and accept. However, the probability that such certificates exist, i.e. the probability that the **AM** protocol for q_i is successful, is at least $1 - 2^{-n}$, conditioned over the random bits Arthur generates at the beginning. The probability, then, that all **AM** protocols are successful, is $\geq (1 - 2^{-n})^{2t(n)}$, which tends to 1 with $n \to \infty$ by Lemma 8 because $2t(n)$ is a polynomial.

Part 2: Soundness. Suppose that M^L rejects. If we feed M^L correct answers to its queries, we reject too. We only feed M^L incorrect answers if one of the **AM** protocols failed. Each **AM** protocol fails with probability $\leq 2^{-n}$, and there are at most $2t(n)$ of them, so with overwhelming probability, all of them succeed. □

We now show that **QAM**∩**co-QAM** is closed under polynomial-time deterministic Turing reductions.

Theorem 14. $\mathbf{P}^{\mathbf{QAM}\cap\mathbf{co\text{-}QAM}} = \mathbf{QAM} \cap \mathbf{co\text{-}QAM}$

Proof. We give a **QAM** protocol for a language $K \in \mathbf{P}^{\mathbf{QAM}\cap\mathbf{co\text{-}QAM}}$. Let M^L be a deterministic Turing machine with oracle access to $L \in \mathbf{QAM} \cap \mathbf{co\text{-}QAM}$, which decides K. In this case there are two uniformly generated quantum circuit families $\{Y_s\}$ and $\{N_s\}$ which answer L and $\overline{L}$, respectively. We exchange random bits and certificates the same way we did in Theorem 13. We expect Merlin to send us (i) the strings that M^L is going to query, (ii) the answers to those queries and (iii) quantum certificates $|\psi\rangle = |\psi_1\rangle\,|\phi_1\rangle \otimes \cdots \otimes |\psi_{t(n)}\rangle\,|\phi_{t(n)}\rangle$ for the these quantum circuits. As in the proof of Theorem 4, we measure the bits that we expect to be classical bits before we execute the **QAM** protocols.

Part 1: Completeness. If M^L accepts, then Merlin will send us the correct answers and unentangled quantum certificates, if they exist, but as noted before, good certificates exist with overwhelming probability. However, even if we give Y_s a good certificate, it may still err because it is a quantum circuit. Fortunately, it is known that **QAM** protocols such as Y_s can be amplified to err with $\leq 2^{-n}$ error, so the previous completeness argument goes through.

Part 2: Soundness. By previous observations, for Arthur to fail it is necessary that at least one **QAM** protocol fails. In the classical case, the soundness of the protocols Y and N was sufficient to reduce the probability that any query failed. The difference in the quantum case is that Merlin can entangle the certificates. But if a quantum circuit Y_s rejects *all* input states with probability $\geq p$, then by the Entanglement Independence Lemma (Lemma 6) it rejects all states regardless of how they are entangled with other qubits that the circuit does not touch, with probability $\geq p$. In our case, $p \geq (1 - 2^{-n})$, so we have soundness by the same argument as in Theorem 13. □

Theorem 15. $\mathbf{NP}^{\mathbf{AM}\cap\mathbf{co\text{-}AM}} \subseteq \mathbf{AM}$.

Proof. We will give an **AM** protocol for a language $A \in \mathbf{NP}^{\mathbf{AM}\cap\mathbf{co\text{-}AM}}$ accepted by nondeterministic Turing Machine M^L. We generate enough random bits, and Merlin responds with certificates to all the queries we are going to make, and with the nondeterministic bits which allegedly make M accept. Since the nondeterministic bits are fixed, the remaining computation is simply a $\mathbf{P}^{\mathbf{AM}\cap\mathbf{co\text{-}AM}}$ computation, which is covered in Theorem 13 □

The following Theorem is the **QAM** analogue of Theorem 10.

Theorem 16. $\mathbf{NP}^{\mathbf{QAM}\cap\text{co-}\mathbf{QAM}} \subseteq \mathbf{QAM}$

Proof. Similar to the previous three theorems, a **QAM** simulation of a language in $\mathbf{NP}^{\mathbf{QAM}\cap\text{co-}\mathbf{QAM}}$ starts with sending enough random bits to Merlin and receiving a quantum state allegedly representing a classical certificate for the nondeterministic machine and quantum certificates so we can simulate the oracle queries. We measure these certificate bits in the computational basis, in addition to the qubits containing the queries and their alleged answers. The nondeterministic input is fixed now, so the remainder of the protocol simulates a deterministic machine making queries to a language in $L \in \mathbf{QAM} \cap \text{co-}\mathbf{QAM}$, which is covered in Theorem 14. □

The **QAM** analogues of the main theorems of this paper follow immediately from Theorem 16, using the techniques used in Section 5.

Theorem 17. *If* $\mathbf{QAM} = \text{co-}\mathbf{QAM}$ *then* $\mathbf{QAM} = \mathbf{PH}^{\mathbf{QAM}}$

Proof. This follows from Theorem 16 by induction. The proof is exactly the same as that of 11, with **QAM** in lieu of **QMA**. We omit the details. □

Theorem 18. *If* $\mathbf{QAM} = \text{co-}\mathbf{QAM}$ *then* $\mathbf{PH} \subseteq \mathbf{QAM}$.

Proof. Clearly $\mathbf{PH} \subseteq \mathbf{PH}^{\mathbf{QAM}}$. By Theorem 17, if $\mathbf{QAM} = \text{co-}\mathbf{QAM}$ then $\mathbf{PH} \subseteq \mathbf{PH}^{\mathbf{QAM}} = \mathbf{QAM}$. □

Theorem 19. *If* $\mathbf{QAM} = \mathbf{PP}$ *then* $\mathbf{PH} \subseteq \mathbf{PP}$. □

7. Discussion and future work

The classical counterpart of Vyalyi's Theorem, obtained by substituting **MA** for **QMA**, is the following consequence of Toda's Theorem.

Theorem 20. *If* $\mathbf{MA} = \mathbf{PP}$ *then* $\mathbf{MA} = \mathbf{CH}$. □

Here **CH** is the counting hierarchy. The consequent of this statement, $\mathbf{MA} = \mathbf{CH}$, is stronger than the consequent we have obtained, which was $\mathbf{QMA} = \mathbf{PH}^{\mathbf{PP}} \subseteq \mathbf{CH}$. Can we get the same result in the quantum case?

Conjecture 1. *If* $\mathbf{QMA} = \mathbf{PP}$ *then* $\mathbf{QMA} = \mathbf{CH}$.

To this end, it suffices to prove what we call a Quantum Toda's Theorem, that **QMA** is low for $\mathbf{P}^{\mathbf{PP}}$:

Conjecture 2 (Quantum Toda's Conjecture)**.** $\mathbf{P}^{\mathbf{PP}^{\mathbf{QMA}}} = \mathbf{P}^{\mathbf{PP}}$. *That is,* **QMA** *is low for* $\mathbf{P}^{\mathbf{PP}}$.

We can see a number of reasons to care about Conjectures 1 and 2. First, Conjecture 1 strengthens Vyalyi's Theorem "to the maximum extent possible", in the sense that it is what happens classically. Second, it would repair a conjecture of Aaronson that if $\mathbf{QMA} \subseteq \mathbf{BQP}_{/\mathrm{qpoly}}$ then $\mathbf{QMA} = \mathbf{CH}$. This appeared as a "theorem" in [9], but he later found an error in the proof [10].

Third, proving Conjecture 2 would establish a quantum version of Toda's Theorem, namely:

$$\mathbf{QMA} \cup \mathbf{QMA}^{\mathbf{QMA}} \cup \mathbf{QMA}^{\mathbf{QMA}^{\mathbf{QMA}}} \cup \cdots \subseteq \mathbf{P}^{\mathbf{PP}}$$

Gharibian et al. recently gave a result in this spirit, proving

Theorem 21 ("Quantum-classical Toda's Theorem", Gharibian et al. [11])**.** $\mathbf{QCPH} \subseteq \mathbf{P}^{\mathbf{PP}^{\mathbf{PP}}}$

Here **QCPH**, defined in [11], is like **PH**, except that the verifier is a uniform family of quantum circuits instead of a deterministic Turing Machine. Proving a Quantum Toda's Conjecture was much of the motivation for the work in this article.

8. Acknowledgements

The first author would like to thank Scott Aaronson, Dorit Aharonov, Harry Buhrman, Mikhail Vyalyi and in particular Ralph Bottesch for many fruitful discussions, and the second author for his supervision of the Master thesis during which the present work was done.

References

[1] Vyalyi, Mikhail N. "QMA=PP implies that PP contains PH." In ECCCTR: Electronic Colloquium on Computational Complexity, technical reports. 2003.

[2] Beigel, Richard. "Perceptrons, PP, and the polynomial hierarchy." Structure in Complexity Theory Conference, 1992., Proceedings of the Seventh Annual. IEEE, 1992.

[3] Gill, John. "Computational complexity of probabilistic Turing machines." SIAM Journal on Computing 6.4 (1977): 675-695.

[4] Toda, Seinosuke. ”PP is as hard as the polynomial-time hierarchy.” SIAM Journal on Computing 20.5 (1991): 865-877.

[5] Arora, Sanjeev, and Boaz Barak. Computational complexity: a modern approach. Cambridge University Press, 2009.

[6] Nielsen, Michael A., and Isaac Chuang. ”Quantum computation and quantum information.” (2002): 558-559.

[7] Kitaev, Alexei Yu, Alexander Shen, and Mikhail N. Vyalyi. Classical and quantum computation. Vol. 47. Providence: American Mathematical Society, 2002.

[8] Marriott, Chris, and John Watrous. ”Quantum Arthur–Merlin games.” Computational Complexity 14.2 (2005): 122-152.

[9] Aaronson, Scott. ”Oracles are subtle but not malicious.” Computational Complexity, 2006. CCC 2006. Twenty-First Annual IEEE Conference on. IEEE, 2005.

[10] Aaronson, Scott. ”Yet more error in papers.” Shtetl-Optimized, 24 May 2017. Retrieved 8 January 2018. https://www.scottaaronson.com/blog/?p=3256

[11] Gharibian, Sevag, et al. ”Quantum generalizations of the polynomial hierarchy with applications to QMA (2).” arXiv preprint arXiv:1805.11139 (2018).

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il