Quantum Complexity, Relativized Worlds, and Oracle Separations

by

Dimitrios I. Myrisiotis

A thesis presented to the Department of Mathematics, of the School of Science, of the National and Kapodistrian University of Athens, in fulfillment of the thesis requirement for the degree of Master of Science in Logic, and the Theory of Algorithms and Computation.

Athens, Greece, 2016

Η Παρούσα Διπλωματική Εργασία
εκπονήθηκε στο πλαίσιο των σπουδών
για την απόκτηση του
Μεταπτυχιακού Διπλώματος Ειδίκευσης
στη
Λογική και Θεωρία Αλγορίθμων και Υπολογισμού
που απονέμει το
Τμήμα Μαθηματικών
του
Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών

Εγκρίθηκε την .... από την Εξεταστική Επιτροπή
αποτελούμενη από τους:

	Ονοματεπώνυμο	Βαθμίδα	Υπογραφή
1.	....	....	....
	....	....	
2.	....	....	....
	....	....	
3.	....	....	....
	....	....	
4.	....	....	....
	....	....	

Notational Conventions

`True`	Denotes the notion of the logical truth.
`False`	Denotes the notion of the logical falsity.
`Yes`	It is used, instead of `True`, to denote the acceptance, of some element, by some algorithm that produces the answer to some decision problem.
`No`	It is used, instead of `False`, to denote the rejection, of some element, by some algorithm that produces the answer to some decision problem.
$\neg P$	Expresses the negation of the logical proposition P, that is, the assertion "it is *not* the case that P."
$P \wedge Q$	Expresses the conjuction of the logical proposition P with the logical proposition Q. It evaluates to `True`, whenever both P and Q are evaluated to `True`. Else, it evaluates to `False`.
$a \in A$	The object a belongs to the set A. The object a can be anything: a number, a set, and so on.
$\{A_i\}_{i \in B}$	Denotes the collection of objects $\{A_i \mid i \in B\}$, where B is a set of indices.
$\bigwedge_i P_i$	Expresses the conjuction of the set of propositions $\{P_i\}_i$. It evaluates to `True`, whenever all of the P_i are evaluated to `True`. Else, it evaluates to `False`.
$P \vee Q$	Expresses the disjunction of the logical proposition P with the logical proposition Q. It evaluates to `True`, whenever either P or Q is evaluated to `True`. Else, it evaluates to `False`.
$\bigvee_i P_i$	Expresses the disjunction of the set of propositions $\{P_i\}_i$. It evaluates to `True`, whenever at least one of the P_i is evaluated to `True`. Else, it evaluates to `False`.
$P \oplus Q$	Expresses the exclusive disjunction, that is, the XOR operation, of the proposition P with the proposition Q. It evaluates to `True` whenever the logical values of P and Q are different. Else, it evaluates to `False`.
$\bigoplus_i P_i$	Expresses the disjunction of the set of propositions $\{P_i\}_i$. It evaluates to `True`, whenever there exist an odd number of propositions $\left\{P_{i_j}\right\}_j$ that evaluate to `True`. Else, it evaluates to `False`.
$P \Rightarrow Q$	Expresses implication, that is, the "the logical proposition P implies the logical proposition Q." It evaluates to `False`, whenever P evaluates to `True` and Q evaluates to `False`. Else, it evaluates to `True`.

$P \Leftrightarrow Q$	Expresses the equivalence of the logical proposition P with the logical proposition Q. It evaluates to `True`, whenever both P and Q are evaluated to the same logical value. Else, it evaluates to `False`.
$\exists$	Denotes the *existential quantifier*.
$\exists!$	Denotes the *existential quantifier*, in a *uniqueness* setting.
$\forall$	Denotes the *universal quantifier*.
$A \cup B$	Denotes the *union* of the sets A and B.
$\bigcup_i A_i$	Denotes the *union* of the family of sets $\{A_i\}_i$.
$A \uplus B$	Denotes the *disjoint union* of the sets A and B.
$\biguplus_i A_i$	Denotes the *disjoint union* of the family of sets $\{A_i\}_i$.
$A \cap B$	Denotes the *intersection* of the sets A and B.
$\bigcap_i A_i$	Denotes the *intersection* of the family of sets $\{A_i\}_i$.
$A \setminus B$	Denotes the *difference* of the set A from the set B.
$A \times B$	Denotes the *Cartesian product* of the sets A and B. That is, the set of all pairs of the form (a, b), for $a \in A$ and $b \in B$.
A^n	For some $n \in \mathbb{N}$, denotes the expression $$\underbrace{A \times A \times \cdots \times A}_{n}.$$ The set A^n contains all of the n-tuples that contain elements from the set A. In the case of strings, the set A^n contains all of the strings of length n, that are composed by objects, that is, "letters," drawn from the set A.
$A^{n \times m}$	Denotes the set of matrices, of size $n \times m$, n rows and m columns, such that their entries are drawn from the set A.
$\|s\|$	Denotes the *length* of the string s. That is, the number of symbols that s contains.
$a \circ b$	Denotes the *concatenation* of the two strings a and b. Often abbreviated to ab.
ϵ	Denotes the *empty* string. That is, for every string s, one has that $$s \circ \epsilon = \epsilon \circ s = s.$$
∞	Denotes a *large-enough* number, yet it is not a number! It is rather an object which we place after the natural numbers. Is it reachable?

A^*	Denotes the quantity $\bigcup_{i=1}^{\infty} A^i.$ That is, it contains all of the finite-length strings made by combining the elements of the set A. The set A^* is called the *Kleene star*.
$\overline{A}$	Denotes the *complement* of the set A. That is, the set $\overline{A} = \{x \mid x \notin A\}.$
$\lvert A \rvert$	Denotes the cardinality of the set A.
$a + b$	Denotes the *addition* operation.
$a - b$	Denotes the *subtraction* operation.
a/b	Denotes the *division* operation, for $b \neq 0$.
$a \cdot b$	Denotes the *multiplication* operation. Often shortened to ab, when it is clear what we mean. We can use it for matrices, numbers, vectors, and such. In the case of vectors, it is called a *dot product*. Often is written as ab.
a^b	Denotes the product $\underbrace{a \cdot a \cdot \ldots \cdot a}_{b},$ for $b \in \mathbb{N}$.
$\lvert z \rvert$	For $z = (a + bj) \in \mathbb{C}$, and $j^2 = -1$, the number $\lvert z \rvert = \sqrt{a^2 + b^2}$ denotes the *magnitude* of z, or the *modulus* of z.
$A = B$	Denotes the relation "the set A is equal to the set B."
$A \subseteq B$	Denotes the relation "the set A is a subset of the set B."
$A \subsetneq B$	Denotes the relation "the set A is a proper subset of the set B."
$\emptyset$	Denotes the empty set. That is, the set that contains no elements.
$\mathbb{N}$	Denotes the set of natural numbers. That is, the set $\{1, 2, \ldots\}.$
$[n]$	Denotes the finite set $\{1, 2, \ldots, n\} \subseteq \mathbb{N}$.

$\mathbb{N}_0$	Denotes the set of natural numbers, augmented with the number zero. That is, $$\mathbb{N}_0 = \mathbb{N} \cup \{0\} = \{0, 1, 2, \dots\}\,.$$
$-A$	Denotes the set that contains the *negatives* of the elements of the set A. That is, $$-A = \{-x \mid x \in A\}\,.$$
$\mathbb{Z}$	Denotes the set of integers. That is, the set $$\{\dots, -2, -1, 0, 1, 2, \dots\} = \mathbb{N}_0 \cup (-\mathbb{N})\,.$$
$\mathbb{Z}_{\text{even}}$	Denotes the set of *even* integers.
$\mathbb{Z}_{\text{odd}}$	Denotes the set of *odd* integers.
$\mathbb{Q}$	Denotes the set of rational numbers. That is, the set $$\left\{\frac{p}{q} \mid p, q \in \mathbb{N}_0 \text{ and } q \neq 0\right\}.$$
$\mathbb{R}$	Denotes the set of real numbers. That is, the points of an infinitely-long, and continuous, straight line.
$[a, b]$	Denotes the closed interval that is bound by the real numbers a and b.
(a, b)	Denotes the open interval that is bound by the real numbers a and b.
$[a, b)$	Denotes the semi-open interval that is bound by the real numbers a and b.
π	Denotes the ratio of the circumreference, of any circle, over its diameter.
e	Denotes *Euler's number*.
$\mathbb{C}$	Denotes the set of complex numbers. That is, the points of an infinitely-long, at its every direction, two-dimensional plane on real axes. Equivalently, one can write that $$\mathbb{C} = \mathbb{R} \times \mathbb{R} = \mathbb{R}^2.$$
$a = b$	Expresses the relation "a is equal to b." The objects a and b can be anything: numbers, functions, matrices, sets, and so on.

$a < b$	Expresses the arithmetic relation "the number a is less than the number b."
$a \ll b$	Expresses the arithmetic relation "the number a is less than the number b, by much." But how much is much? This is defined according to the current context.
$a \leq b$	Expresses the arithmetic relation "the number a is less than, or equal to, the number b."
$a \not\square b$	Expresses the logical negation "it is *not* the case that $a \square b$." It is similar to $\neg P$, but different in the sense that is typeset over relational or logical symbols $\square$. For example, the logical proposition "$a \neq b$" means "a is *not* equal to b." If its use is found to be confusing, one can use the more formal symbolism $\neg P$, instead.
I	Denotes the *identity* transformation.
$M_{i,j}$	Denotes the (i, j) entry of the matrix M. Can be found as $M(i, j)$, too.
M^T	Denotes the *transpose* of the matrix M.
M^*	Denotes the *complex-conjugate* of the matrix M.
z^*	Denotes the *complex-conjugate* $z^* = a - bj$, of the complex number $z = a + bj$. Here, $j^2 = -1$.
$M^\dagger$	Denotes the *conjugate transpose* of the matrix M. That is, $$M^\dagger = (M^*)^T .$$
M^{-1}	Denotes the *inverse* of the matrix M.
$\det(M)$	Denotes the *determinant* of the square matrix M.
$\mathrm{Tr}(M)$	Denotes the *trace* of the square matrix M. That is, the sum of the main diagonal elements of M.
$\mathrm{rank}(M)$	Denotes the *rank* of the matrix M.
$A \otimes B$	Denotes the *Kronecker product* of the matrices A and B. This product contains all the possible products among the pairs of numbers that are such that their first element is in the matrix A, and their second element is in the matrix B. If one of the matrices A and B is a vector, then it is called a *tensor product*.
$\log a$	Denotes the *logarithm* of a, taken to base two.
$\sin\theta$	Denotes the *sine* function, for $\theta \in \mathbb{R}$.
$\cos\theta$	Denotes the *cosine* function, for $\theta \in \mathbb{R}$.
$\sum_{i=b}^{c} a_i$	Denotes the finite sum $a_b + a_{b+1} + \cdots + a_c$, for $b, c \in \mathbb{N}$.
$\prod_{i=b}^{c} a_i$	Denotes the finite product $a_b \cdot a_{b+1} \cdot \ldots \cdot a_c$, for $b, c \in \mathbb{N}$.

$[A, B]_-$	Denotes the *commutator*. That is, the quantity $AB - BA$.
$[A, B]_+$	Denotes the *anti-commutator*. That is, the quantity $AB + BA$.
$\vert\psi\rangle$	Denotes a "ket." That is, a *vector* in Dirac notation. This vector is considered to live in some complex vector space $\mathcal{H}$.
$\langle\psi\vert$	Denotes a "bra." That is, the *dual*, or *conjugate transpose*, of a ket $\vert\psi\rangle$. One has that $$\left(\vert\psi\rangle\right)^{\dagger} = \langle\psi\vert\,,$$ and the vector $\langle\psi\vert$ lives in the space $\mathcal{H}^{\dagger}$, which denotes the *dual* space of $\mathcal{H}$.
$\Vert\,\vert\psi\rangle\Vert_p$	For $p \in \mathbb{N}$, denotes the p-th norm of the ket $\vert\psi\rangle$.
$\mathcal{H}_1 \otimes \mathcal{H}_2$	Denotes the combined space that results from joining the two vector spaces, namely $\mathcal{H}_1$ and $\mathcal{H}_2$, into a common combined space.
$\Pr_{r\sim\mu}\left[E\left(r\right)\right]$	Denotes the probability that the event $E\left(r\right)$ occurs, over the choices of r from the probability distribution μ.
$\mathbb{E}_{r\sim\mu}\left[X\left(r\right)\right]$	Denotes the expectation of the random variable $X\left(r\right)$, over the choices of r from the probability distribution μ.
$f : A \to B$	Denotes the total function f, with domain set A and range set $R \subseteq B$.
$\left(A \to B\right)$	Denotes the set of functions from A to B.
$\mathcal{O}\left(f\left(n\right)\right)$	Denotes the set of functions g, such that there are positive numbers $c > 0$, and $n_0 \in \mathbb{N}$, according to which $$\forall n \geq n_0 : g\left(n\right) \leq cf\left(n\right).$$
$o\left(f\left(n\right)\right)$	Denotes the set of functions g, such that for all positive numbers $c > 0$, there exists some $n_0 \in \mathbb{N}$, according to which $$\forall n \geq n_0 : g\left(n\right) < cf\left(n\right).$$
$\Omega\left(f\left(n\right)\right)$	Denotes the set of functions g, such that there are positive numbers $c > 0$, and $n_0 \in \mathbb{N}$, according to which $$\forall n \geq n_0 : g\left(n\right) \geq cf\left(n\right).$$

$\omega(f(n))$	Denotes the set of functions g, such that for all positive numbers $c > 0$, there exists some $n_0 \in \mathbb{N}$, according to which $$\forall n \geq n_0 : g(n) > cf(n).$$
$\Theta(f(n))$	Denotes the set of functions g, such that $g(n) \in \mathcal{O}(f(n))$, and $g(n) \in \Omega(f(n))$. That is, $$\Theta(f(n)) = \Omega(f(n)) \cap \mathcal{O}(f(n)).$$
$\delta_{i,j}$	Denotes the *Kronecker delta* function. That is, $$\delta_{i,j} = \delta(i,j) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$
CLASS	Denotes some computational complexity class, namely CLASS. Note the UPPERCASE $\mathsf{sans\ serif}$ font.
$\mathsf{CLASS}^{\mathcal{A}}$	Denotes some computational complexity class, namely CLASS, that has access to some oracle $\mathcal{A}$.
$\mathsf{CLASS}^{\mathcal{A}[k(n)]}$	Denotes some computational complexity class, namely CLASS, that has access to some oracle $\mathcal{A}$, for only $k(\lvert x \rvert) \in \mathbb{N}$ queries, nonetheless, where x is the input to some problem L in the class $\mathsf{CLASS}^{\mathcal{A}}$.
`PROBLEM`	Denotes the computational problem `PROBLEM`. Note the UPPERCASE `typewriter` font.
SET	Denotes a set with some special property, that is, a specific set that contains some similar elements. For example, ORACLES denotes the set of all oracles, be them classical or quantum ones.

Abstract

The complexity class QMA, defined by Watrous, in 2000, is the quantum analogue of MA, defined by Babai, in 1985, which, in turn, is a generalization of the class NP. The class MA generalizes the class NP in the sense that the verification procedure of the purported proof, put forth by the prover, is carried out by a probabilistic machine, rather than a deterministic one—as the definition of the class NP demands.

In 2014, Grilo, Kerenidis, and Sikora, proved that the quantum proof, in the setting of QMA, may always be replaced by, an appropriately defined, quantum subset state—without any conceptual loss. That is, $\mathsf{QMA} \subseteq \mathsf{SQMA}$. Grilo et al., named their new class SQMA, for *subset-state quantum Merlin-Arthur*. Thus, one could write that $\mathsf{SQMA} = \mathsf{QMA}$, as the inclusion $\mathsf{SQMA} \subseteq \mathsf{QMA}$ holds trivially.

After this result, by Grilo, Kerenidis, and Sikora, Fefferman and Kimmel, in 2015, used this new characterization of QMA, and further proved that there exists some quantum oracle $\mathcal{A}$—similar to that Aaronson and Kuperberg introduced, and used, in 2006, to show that $\mathsf{QMA}_1^{\mathcal{A}} \not\subseteq \mathsf{QCMA}^{\mathcal{A}}$—which is such that $\mathsf{QMA}^{\mathcal{A}} = \mathsf{SQMA}^{\mathcal{A}} \not\subseteq \mathsf{QCMA}^{\mathcal{A}}$. Here, QCMA is that version of QMA, defined by Aharonov, and Naveh, in 2002, in which the purported proof is purely-classical, that is, a bitstring, and QMA_1 is the *perfect completeness* version of QMA. In their separation, Fefferman and Kimmel introduced, and used, an interesting template to obtain oracle separations against the class QCMA.

Drawing upon this recent result, by Fefferman and Kimmel, we prove that there exists some quantum oracle $\mathcal{A}$, such that $\mathsf{SQMA}_1^{\mathcal{A}} \not\subseteq \mathsf{QCMA}^{\mathcal{A}}$. We note that the class SQMA_1 is the *perfect completeness* version of the class SQMA. In our proof, we used the template of Fefferman and Kimmel, a modified version of their basic quantum oracle construction, as well as the basic decision problem, that they themselves used for their separation. Note that our result implies that of Fefferman and Kimmel, as the inclusion

$\mathsf{SQMA}_1 \subseteq \mathsf{SQMA}$ holds.

After we state and prove our result, we take a detour to explore a bit the world of oracle separations, both in the classical and the quantum setting. That is, we explore some results, and their underlying methods, about classical and quantum oracles being employed for proving separations—about classical, or quantum, complexity classes. Hence, we investigate some gems pertaining to the, not few at all, nor uninteresting, privileged relativized worlds.

Finally, we return, to the research setting, to approach the open question of whether there exists some classical, or quantum, oracle $\mathcal{A}$, such that $\mathsf{QMA}_1^{\mathcal{A}} \not\subseteq \mathsf{SQMA}_1^{\mathcal{A}}$, or not. We record our efforts, and some of our first ideas, thus far.

Keywords. Quantum computational complexity theory, computational complexity theory, relativized worlds, oracle separations, oracles, QMA, QCMA, SQMA, SQMA_1, quantum states, quantum subset-states, quantum proofs, verification procedures, verification protocols, diagonalization, and query complexity.

Acknowledgements

I would like to thank all the little people who made this possible.[1] As it turns out, they are quite many.

First, I would like to thank, my unofficial advisor, yet official member of my thesis-committee, Iordanis Kerenidis. Professor Kerenidis gave me the opportunity to do research in quantum complexity theory, to explore some of its most fascinating aspects, and to produce results, that would be unattainbale, for me, without his assistance, guidance, and support. Along these lines, I owe a lot to the Ph.D. candidate, of Université Paris Diderot, Alex Grilo, for helping me grasp some of the most advanced concepts of quantum complexity theory, and for wholeheartedly responding to my many, and often ambiguous, and annoying, questions. It was on May 2016, that I visited Professor Kerenidis at Université Paris Diderot, and, together with Alex Grilo, we worked on the SQMA_1 versus QCMA problem, and on the QMA_1 versus SQMA_1 problem. At the end, together, we managed to somewhat resolve, in an alternative universe, the SQMA_1 versus QCMA problem. I am very, very, happy, and thankful, about our collaboration, which was the fertile ground for the Chapters 5 and 6 to grow upon.

Regarding the other three members of my thesis-committee, Efstathios Zachos, Dimitrios Fotakis, and Aristeidis Pagourtzis, I should admit that I am at a loss of words. Really. Where should I start? What should I put forth first? I met Professor Emeritus Zachos, who is, by the way, my official advisor, Assistant Professor Fotakis, and Associate Professor Pagourtzis, back in 2011, when I was an undergraduate mechanical engineering student at NTUA. They were very friendly, and, above all, very, very, encouraging for me to pursue my graduate studies in theoretical computer science. Since then, their helpful presence has become evident, through their graduate courses, the class projects, the talks, and so on. They were just there: reachable, willing to help.

Now, nothing of all these would have taken place, if it were not for my parents. My parents. Two people: always there to encourage me to set goals, and go after them. To not give up, to persevere. My parents. Their sacrifices, all these years, and their love, is a most valuable resource for me. Their sincerity, their patience. My parents.

Finally, I owe a big “Thank you!” to Professor Evangelos Papadopoulos, my

[1] This truthful sentence was copied from the, available to the public, template about the preparation of theses documents, that is created by the people of the University of Waterloo, and subsequently pasted here.

mechanical engineering advisor, and to the Ph.D. candidate, of the NTUA School of Mechanical Engineering, Ioannis Kontolatis, for helping me in making my first steps in academic research, to Nefeli Antoulinaki, for unawaringly helping me realize that I would really enjoy quantum computing, as the broader area of my thesis, while in Corfu, in October 2015, to Christoforos Mavrogiannis, for all the encouraging quantum Skype-calls, over networks of questionable quality, and to Kostantinos Mpraimakis, for all our quantum-flavored conversations during our late-afternoon coffee-meetings in NTUA. I also owe many thanks to Georgios Karlis, Ioannis Choustoulakis, Loukas Mpafatakis, Dimitrios Agas, Emmanouil Panagiotopoulos, and Konstantinos Kentrirloglou, for being awesome, imaginative, and very supportive super-friends.

This thesis was typeset using LaTeX's `mathpazo` package, and the quantum circuits were drawn using the `qcircuit` package. We also used, in many parts of this thesis, the M.Sc.-Ph.D. thesis template that is created, and kindly freely-distributed, by the people of the University of Waterloo.

Στους αγαπημένους μου γονείς, Ηλία και Πόπη.

To my beloved parents, Popi and Ilias.

Table of Contents

List of Algorithms

List of Figures

List of Tables

Chapter 1

Introduction

Please read this section carefully. Believe me, it will pay off.

—Leonard Susskind, and Art Friedman,
Quantum Mechanics (2014)

In this chapter, we introduce the reader to *Quantum Computing Theory*, and to *Quantum Computational Complexity Theory*. We also point out what is to come, next, in the subsequent chapters of this thesis.

1.1 Classical Computational Complexity Theory

Computational complexity theory is the field of computing that is concerned about quantifying the efficiency of algorithms, and categorizing the computational problems in classes, according to their computational hardness. This field saw many changes in the past years, which led to the formation of new theories and scientific branches. However, all of these new theories are somewhat contained in our world: they are all classical, in the sense that their whole mechanism can be explained within classical physics. Thus, turning to quantum mechanics was a very refreshing, and promising, perspective for this area.

1.2 Quantum Computational Complexity Theory

Quantum computing is what one gets if one mixes up computer science and quantum mechanics. That is, we exploit some weird, and counterintuitive, quantum-mechanical phenomena that, at the end, enable us two devise algorithm that have no classical, until now, counterpart, in terms of efficiency.

To understand what a quantum computer is, we will compare it to the two classical paradigms of realistic, and physically realizable, computers, namely the deterministic computers, and the probabilistic ones. Suppose that we have a 16-register computer, in which each register assumes the value of a bit. Clearly, this computer has 2^{16} configurations, see Figure 1.1.

$$\underbrace{\Box\,\Box\,\Box\,\Box\,\Box\,\Box\,\Box\,\Box\,\Box\,\Box\,\Box\,\Box\,\Box\,\Box\,\Box\,\Box}_{\text{16 registers}}$$

$$\forall \Box : \Box \in \{0,1\}$$

Figure 1.1: Our 16-bit computer, with 2^{16} configurations.

Now, we ask the question *"how many numbers are necessary for one to communicate the state of our 16-bit computer?"* It depends. In the case that our computer is deterministic, we clearly need to communicate 16 numbers: the value of each bit, namely b_i, for every $i \in [16]$, see Figure 1.2.

$$b_1\ b_2\ b_3\ b_4\ b_5\ b_6\ b_7\ b_8\ b_9\ b_{10}\ b_{11}\ b_{12}\ b_{13}\ b_{14}\ b_{15}\ b_{16}$$

$$\forall i \in \{1,2,\ldots,16\} : b_i \in \{0,1\}$$

Figure 1.2: Communicating a configuration of a deterministic 16-bit computer.

However, in the case that we consider the computer configurations to be lexicographically ordered, we need only communicate a number. That is, the number that corresponds to the index of the configuration at hand. In Figure 1.3a, we see that one needs to send 2^{16} numbers, of which only one is non-zero: the one that corresponds to the current configuration of our computer.

What happens in the case that our computer is probabilistic? In this case, one needs to communicate a probability distribution over all of the possible 2^{16} configurations. Thus, in this case, one needs to communicate 2^{16} numbers, in $[0, 1]$, that sum to 1, see Figure 1.3b.

What happens in the case that our computer is quantum? In this case, one needs to communicate 2^{16} complex numbers! That is, we need one complex number for each configuration of our computer. Our only demand, here, is that these complex numbers are such that the sum of the squares of their magnitudes sum to 1, see Figure 1.3c.

As one observes, probabilistic computers can be seen as a generalization of deterministic computers, and, in turn, quantum computers can be seen as a generalization of probabilistic ones.

In order to further explain what quantum computers are, we shall write a few more words. Until now, we know that in order to communicate the configuration of an n-qubit quantum computer, we need to transmit 2^n complex numbers: one for each of its configurations. If we create a basis vector state for each of the configurations of our quantum machine, we get that every state of a quantum computer can be sought as an exponentially big vector, with complex entries, as the state of such a quantum machine is the weighted sum of these basis vector states. The respective coefficients are the 2^n complex numbers we need to send to communicate its state. As we are going to see, in Chapter 2, these complex vectors evolve in a unitary fashion, that is, any two successive states, of a quantum computer, are such that the later is obtained by applying a unitary transformation on the former. In order to access them, we perform measurements, which, for a system of dimension 2^n, for some natural $n \in \mathbb{N}$, we get only n bits of information.

Quantum algorithms seem to be more efficient because they make use of the feature of interference between the possible computational paths, in order to destructively combine paths that lead to a wrong answer, and to creatively combine the paths that lead to the right one. Note that interference is only possible in the quantum setting, since, even in the probabilistic setting, which is perhaps the most powerful, in a classical sense, all the quantities that there emerge, as probabilities, are positive. This means that these quantities, when combined, add up to bigger probabilities. In the quantum setting, however, since we allow for negative complex amplitudes, sometimes the mixing leads to mutual cancellation, or mutual strengthening, of paths, and so on. Thus, we are able to destroy bad computational paths, which lead to the wrong answer, and to strengthen the good computational paths, which lead to the correct answer. Note that there exist bad computational paths, since we allow for our quantum computers to err, occasionally. Allowing for errors is not a new feature: probabilistic computers do have that feature, too. The whole point is to contain the error probability in some

$$p_1, p_2, \ldots, p_{2^{16}}$$

$$\forall i \in \left\{1, 2, \ldots, 2^{16}\right\} : p_i \in \{0, 1\}$$

$$\sum_{i=1}^{2^{16}} p_i = 1 \Leftrightarrow \exists! k \in \left\{1, 2, \ldots, 2^{16}\right\} : p_k = 1$$

(a) Communicating a configuration of a deterministic 16-bit computer. Note that we need to communicate only a number.

$$p_1, p_2, \ldots, p_{2^{16}}$$

$$\forall i \in \left\{1, 2, \ldots, 2^{16}\right\} : p_i \in [0, 1]$$

$$\sum_{i=1}^{2^{16}} p_i = 1$$

(b) Communicating a configuration of a probabilistic 16-bit computer. Note that we need to communicate 2^{16} numbers.

$$c_1, c_2, \ldots, c_{2^{16}}$$

$$\forall i \in \left\{1, 2, \ldots, 2^{16}\right\} : c_i \in \mathbb{C}$$

$$\sum_{i=1}^{2^{16}} |c_i|^2 = 1$$

(c) Communicating a configuration of a quantum 16-bit computer. Note that we need to communicate 2^{16} numbers.

Figure 1.3: A comparative treatment of deterministic, probabilistic, and quantum, 16-register computers.

reasonable range, thus leading to bouned-error computational complexity classes like BPP and BQP.

1.3 Complexity Classes

Complexity classes, typeset as CLASS, are classes of sets, that is, classes of languages. These languages encode the `Yes`-instances of various decision problems, typeset as `PROBLEMS`, like `REACHABILITY`, which, more or less, asks if there is a path between two given vertices of a given graph.

For example, we use P to denote the class of the decidable languages that can be decided by polynomial-time deterministic classical computers, and NP to denote the class of languages that can be decided by polynomial-time non-deterministic computers. Many famous questions are expressed in this context, like the P versus NP question, that is,

$$\mathsf{P} \overset{?}{=} \mathsf{NP}. \tag{1.1}$$

1.4 Relativized Worlds

Suppose that there is a fictional imaginary world where we can solve any instance of, say, `SAT` in constant time. What would that imply? Many interesting things, as we are going to see. These imaginary constructions are called *relativized worlds*. The ability to solve a fixed problem in constant time, by a hypothetical super-algorithm, which is called an *oracle*, enables us to prove things inconceivable before.

In Figure 1.4, one can inspect these worlds. By W we denote our world, and, by $\mathsf{W}^{\mathcal{A}}$, the relativized one. Note that, in the general case, their intersection is non-empty, while none is a superset of the other.

1.4.1 Oracle Collapses

By using oracles, one can make a big complexity class coincide with one of its subsets.

Theorem 1.1 ([80])**.** There is some oracle, namely $\mathcal{A}$, such that

$$\mathsf{P}^{\mathcal{A}} = \mathsf{NP}^{\mathcal{A}}. \tag{1.2}$$

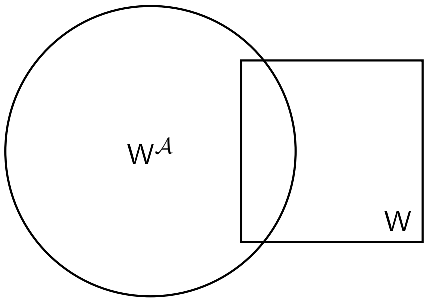

Figure 1.4: Our world, namely W, and a relativized world $W^{\mathcal{A}}$, induced by calls to some oracle $\mathcal{A}$. In this Venn diagram, we depict the relationship between the corresponding sets that hold all of the truthful logical propositions of each world.

Proof. Let $\mathcal{A}$ be the language that encodes all of the `Yes`-instances of the problem `QSAT`, defined, below, in Table 1.1. Note that the problem `QSAT` is a PSPACE-complete one [80, 87].

Table 1.1: The problem `QSAT`.

`QUANTIFIED SATISFIABILITY` (`QSAT`)	
Input	A quantified Boolean formula ϕ.
Output	A `Yes` or a `No` reply, regarding whether the formula ϕ is satisfiable, or not, respectively.

For

$$\mathcal{A} = \text{the language, or set, that encodes all of the Yes-instances of QSAT,} \tag{1.3}$$

we have that

$$\begin{aligned} \mathsf{NP}^{\mathcal{A}} &\subseteq_{(1)} \mathsf{NPSPACE} \\ &\subseteq_{(2)} \mathsf{PSPACE} \\ &\subseteq_{(3)} \mathsf{P}^{\mathcal{A}}. \end{aligned} \tag{1.4}$$

Thus, we get the inclusion $\mathsf{NP}^{\mathcal{A}} \subseteq \mathsf{P}^{\mathcal{A}}$. Since one has that $\mathsf{P}^{\mathcal{A}} \subseteq \mathsf{NP}^{\mathcal{A}}$, for every possible oracle $\mathcal{A}$, one has that $\mathsf{P}^{\mathcal{A}} = \mathsf{NP}^{\mathcal{A}}$. We will now prove each of the three inclusions of the Equation (1.4), separately.

$\subseteq_{(1)}$: The languages of the class $\mathsf{NP}^{\mathcal{A}}$ can be decided in NPSPACE, by replacing the oracle for `QSAT` by an open, white-box-natured, algorithm that requires polynomial space to provide us with a solution to any given instance of `QSAT`. Such an algorithm exists, since `QSAT` is in PSPACE.

$\subseteq_{(2)}$: The languages of the class NPSPACE can decided by PSPACE-machines, as a consequence of Savitch's theorem [80, 87].

$\subseteq_{(3)}$: The class $\mathsf{P}^{\mathcal{A}}$ can decide all of the languages in the class PSPACE, since the class $\mathsf{P}^{\mathcal{A}}$ has access to an oracle $\mathcal{A}$, which is the PSPACE-complete decision problem `QSAT`.

□

1.4.2 Oracle Separations

It turns out that there is a more intriguing use of oracles: the one that is about separating classes from each other. While is almost trivial to impose an oracle identification between any two complexity classes, it is a lot more challenging and rewarding to devise an oracle separation between two given classes of interest. A first original example of such a work is the brilliant result by Baker, Gill, and Solovay [96]. According to this result, there is a classical oracle $\mathcal{A}$, relative to which

$$\mathsf{NP}^{\mathcal{A}} \not\subseteq \mathsf{P}^{\mathcal{A}}. \tag{1.5}$$

That is, they showed that there exists some language L, that is in $\mathsf{NP}^{\mathcal{A}}$, yet not in $\mathsf{P}^{\mathcal{A}}$. While many may contend that these results are not that useful, they constitute a very interesting conceptual mechanism for exploring some aspects of computational complexity that are left untouched by the standard separating techniques.

1.5 About this Thesis

In this thesis we seek to better our understanding about the relationship of the class SQMA_1, and some of its relative classes such as QMA_1 and QCMA, see the Subsection 2.4.2. In particular, by using a recent seminal result [27] by Fefferman and Kimmel, we show that there exists a relativized world, induced by calls to some appropriate oracle $\mathcal{A}$, such that

$$\mathsf{SQMA}_1^{\mathcal{A}} \not\subseteq \mathsf{QCMA}^{\mathcal{A}}. \tag{1.6}$$

We also present some first attempts towards separating, with the use of some oracle $\mathcal{A}$, the class $\mathsf{QMA}_1^{\mathcal{A}}$ from $\mathsf{SQMA}_1^{\mathcal{A}}$, that is, towards

$$\mathsf{QMA}_1^{\mathcal{A}} \not\subseteq \mathsf{SQMA}^{\mathcal{A}}. \tag{1.7}$$

1.6 Thesis Structure

The rest of this thesis is organized as follows.

- In Chapter 2, we present the mathematical preliminaries that are necessary for someone to grasp the, more advanced, concepts that follow in the subsequent chapters.
- In Chapter 3, we survey the literature about oracle separations, both in the classical and the quantum setting.
- In Chapter 4, we review the basic methods and paradigms, introduced so far, in the oracle separation literature, and lie close to our results from an epistemological point of view.
- In Chapter 5, we present our main results, along with detailed proofs.
- In Chapter 6, we discuss, with the reader, the consequences, as well as the possible interpretations, of our results. Finally, we conclude this thesis, review our contribution, and state some open problems, while outlining some legitimate future work directions.

Chapter 2

Preliminaries

We assume the reader has a strong background in elementary linear algebra.

—Phillip Kaye, Raymond Laflamme, and Michele Mosca,
An Introduction to Quantum Computing (2007)

In this chapter, we introduce the reader to the fundamental concepts that form the conceptual background of this thesis.

2.1 Complex Analysis

We denote by $\mathbb{C}$ the set of complex numbers, that is, the points of an infinitely-long, at its every direction, two-dimensional plane on real axes. A typical form of a complex number, for real numbers a and b, and $j^2 = -1$, is

$$c = a + bj. \tag{2.1}$$

The form of the Equation (2.1), is called the *Cartesian form* of the complex number c. Now, to complete our understanding about the complex numbers, we need some more notions, portrayed in the Definitions 2.1, 2.2, and 2.3.

Definition 2.1 (The Complex-Conjugate of a Complex Number)**.** Let $j^2 = -1$. Let, also, the quantity $c = (a + bj) \in \mathbb{C}$ be a complex number. We define the *complex-conjugate* of c, namely c^*, to be the complex number

$$c^* = a - bj. \tag{2.2}$$

Definition 2.2 (The Magnitude of a Complex Number)**.** Let $j^2 = -1$. Let, also, the quantity $c = (a + bj) \in \mathbb{C}$ be a complex number. We define the *magnitude*, or the *modulus*, of c, namely $|c|$, to be the real number

$$\begin{aligned} |c| &= \sqrt{a^2 + b^2} \\ &= \sqrt{c^* c}. \end{aligned} \tag{2.3}$$

That is,

$$|c|^2 = c^* c = c c^*. \tag{2.4}$$

Definition 2.3 (Polar Form of a Complex Number)**.** Let $j^2 = -1$. The *polar form* of some complex number

$$c = a + bj \in \mathbb{C}, \tag{2.5}$$

is

$$c = r\, e^{\phi j}, \tag{2.6}$$

with

$$\begin{aligned} r &= |z| \\ &= \sqrt{a^2 + b^2}, \end{aligned} \tag{2.7}$$

and

$$\phi = \arctan\left(\frac{b}{a}\right). \tag{2.8}$$

Definition 2.4 (Phase Factors)**.** A *phase factor* is a complex number with magnitude equal to one. That is, the complex number $c \in \mathbb{C}$ is a phase factor if, and only if, $|c| = 1$. Note that, if

$$c = a + bj \in \mathbb{C}, \tag{2.9}$$

then, for $r = |c| = 1$, we have that

$$\begin{aligned} c &= r\, e^{\phi j} \\ &= e^{\phi j}, \end{aligned} \tag{2.10}$$

with

$$\phi = \arctan\left(\frac{b}{a}\right). \tag{2.11}$$

2.2 Linear Algebra

Quantum computing is largely based on linear algebra, so, in this section, we will provide the reader with enough information about some important notions and theorems drawn from this discipline.

What is a vector? Well, it is considered to be a collection of numbers, like an ordered tuple, for example, usually drawn from a large set as $\mathbb{C}$. That is, vectors live in spaces like $\mathbb{C}^N$, where N denotes the number of the numbers the afore-mentioned tuple contains, or, equivalently, the dimension of the vectors that we consider. In this work, we will consider vector spaces of exponentially-big, in n, dimension $N = 2^n$, where $n \in \mathbb{N}$ is some natural number. We assume that the reader is well-acquainted with the notion of a vector space.

Since, in this work, we are interested in studying quantum computing, and such, we are going to use the Dirac notation for vectors, and, other, similar in flavor, objects.

Definition 2.5 (Kets and Bras)**.** A *ket*, namely $|\psi\rangle$, is an alternative way to represent a vector. That is,

$$\begin{aligned} |\psi\rangle &= \vec{\psi} \\ &= \boldsymbol{\psi} \\ &= \text{a vector.} \end{aligned} \tag{2.12}$$

A *bra*, namely $\langle\psi|$, is an alternative way to represent the *dual*, that is, the conjugate transpose, of some vector, or ket, $|\psi\rangle$. That is,

$$\begin{aligned} \langle\psi| &= \left(\vec{\psi}\right)^\dagger \\ &= \boldsymbol{\psi}^\dagger \\ &= \text{the conjugate transpose of some vector } |\psi\rangle \\ &= \text{the dual of some vector } |\psi\rangle \\ &= \left(|\psi\rangle\right)^\dagger . \end{aligned} \tag{2.13}$$

Definition 2.6 (The L_p-norm)**.** Let $p, n \in \mathbb{N}$ be natural numbers. We have that the L_p-norm, of some vector

$$x = (x_1, x_2, \ldots, x_n) \in \mathbb{C}^n, \tag{2.14}$$

is given by the equation

$$L_p(x) = \|x\|_p$$

$$= \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}. \tag{2.15}$$

Note 2.7. Note that the L_1-norm is just the sum of the magnitudes of the components of the input vector x, that is,

$$\begin{aligned} L_1(x) &= \|x\|_1 \\ &= \sum_{i=1}^{n} |x_i| \\ &= |x_1| + |x_2| + \cdots + |x_n|. \end{aligned} \tag{2.16}$$

In a similar manner, one observes that the L_2-norm is just the square root of the sum of the squares of the magnitudes of the components of the input vector x, that is,

$$\begin{aligned} L_2(x) &= \|x\|_2 \\ &= \sqrt{\sum_{i=1}^{n} |x_i|^2} \\ &= \sqrt{|x_1|^2 + |x_2|^2 + \cdots + |x_n|^2}. \end{aligned} \tag{2.17}$$

Definition 2.8 (Linear Transformations [91]). A linear transformation on a vector space $\mathcal{H}$ is a function

$$T : \mathcal{H} \to \mathcal{H}, \tag{2.18}$$

such that

$$\forall |\psi\rangle, |\phi\rangle \in \mathcal{H} : T(|\psi\rangle + |\phi\rangle) = T(|\psi\rangle) + T(|\phi\rangle), \tag{2.19}$$

and

$$\forall |\psi\rangle \in \mathcal{H}, \forall \alpha \in \mathbb{R} : T(\alpha |\psi\rangle) = \alpha T(|\psi\rangle). \tag{2.20}$$

Definition 2.9 (Types of Linear Transformations). We say that a linear transformation T is *unitary*, if $T^{-1} = T^{\dagger}$, that is, if the inverse T^{-1} is equal to the transpose conjugate $T^{\dagger}$. A linear transformation is *Hermitean*, if $A = A^{\dagger}$. Finally, a linear transformation is *normal*, if

$$AA^{\dagger} = A^{\dagger}A, \tag{2.21}$$

or, equivalently,

$$\left[A, A^{\dagger}\right]_{-} = 0. \tag{2.22}$$

Theorem 2.10 (The Spectral Theorem)**.** Any normal linear transformation can be decomposed as a linear combination of some outer products, that are created by appropriately combining its eigenvectors. The coefficients of these linear combinations are the eigenvalues of the transformation.

To be more precise, for every linear transformation T, acting on a finite-dimensional vector space $\mathcal{H}$, there is an orthonormal basis of $\mathcal{H}$, consisting of the eigenvectors $|T_i\rangle$, which correspond to the eigenvalues T_i, of T. This implies that

$$T = \sum_i T_i \, |T_i\rangle \langle T_i| \, . \tag{2.23}$$

Definition 2.11 (Linear Independence)**.** Two objects, say, a and b, are *linearly independent*, whenever, for $\lambda_1, \lambda_2 \in \mathbb{R}$,

$$\lambda_1 a + \lambda_2 b = 0 \Rightarrow \lambda_1 = \lambda_2 = 0. \tag{2.24}$$

Definition 2.12 (Rank of a Matrix)**.** By *rank* of some matrix M, denoted as $\operatorname{rank}(M)$, we mean the maximum number of linearly independent rows, or columns, of M.

Definition 2.13 (Trace of a Matrix)**.** The *trace* of some square matrix M, of size $n \times n$, for some $n \in \mathbb{N}$, denoted as $\operatorname{Tr}(M)$, is the sum of its main diagonal elements. That is,

$$\operatorname{Tr}(M) = \sum_{i=1}^{n} M_{ii}. \tag{2.25}$$

Note 2.14 (Cyclical Property of Trace)**.** Note that, for two matrices, namely M_1 and M_2, one has that

$$\operatorname{Tr}(M_1 M_2) = \operatorname{Tr}(M_2 M_1) \, . \tag{2.26}$$

Definition 2.15 (Kronecker Product of Matrices)**.** For

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathbb{C}^{2\times 2}, \tag{2.27}$$

and

$$B = \begin{pmatrix} p & q \\ r & s \end{pmatrix} \in \mathbb{C}^{2\times 2}, \tag{2.28}$$

we have that the *Kronecker product of the matrices A and B* is

$$\begin{aligned} A \otimes B &= \begin{pmatrix} aB & bB \\ cB & dB \end{pmatrix} \\ &= \begin{pmatrix} a\begin{pmatrix} p & q \\ r & s \end{pmatrix} & b\begin{pmatrix} p & q \\ r & s \end{pmatrix} \\ c\begin{pmatrix} p & q \\ r & s \end{pmatrix} & d\begin{pmatrix} p & q \\ r & s \end{pmatrix} \end{pmatrix} \\ &= \begin{pmatrix} ap & aq & bp & bq \\ ar & as & br & bs \\ cp & cq & dp & dq \\ cr & cs & dr & ds \end{pmatrix} \in \mathbb{C}^{4\times 4}. \end{aligned} \tag{2.29}$$

Remark 2.16. Note that, for matrices A and B, that is, linear transformations in matrix form, and states $|\psi\rangle$ and $|\phi\rangle$, one gets

$$(A \otimes B)\left(|\psi\rangle \otimes |\phi\rangle\right) = \left(A\,|\psi\rangle\right) \otimes \left(B\,|\phi\rangle\right). \tag{2.30}$$

Definition 2.17 (Kronecker Product of Vector Spaces)**.** For two vector spaces, namely $\mathcal{A}$ and $\mathcal{B}$, with bases $\{|a_i\rangle\}_{i=1}^{n_a}$ and $\{|b_j\rangle\}_{j=1}^{n_b}$, respectively, for naturals n_a and n_b, we have that

$$\begin{aligned} \mathcal{A} \otimes \mathcal{B} = \Bigg\{ \sum_{i\in[n_a]} \sum_{j\in[n_b]} \lambda_{ij}\,|a_i\rangle \otimes |b_j\rangle \mid \lambda_{ij} \in \mathbb{C} \\ \text{and} \sum_{i\in[n_a]} \sum_{j\in[n_b]} \left|\lambda_{ij}\right|^2 = 1 \Bigg\}. \end{aligned} \tag{2.31}$$

That is, $\mathcal{A} \otimes \mathcal{B}$ contains all of the linear combinations of all the possible Kronecker products of the bases of $\mathcal{A}$ and $\mathcal{B}$.

2.3 Quantum Mechanics

Quantum Mechanics is that part of *Physics* which accurately describes very small physical systems. For example, quantum mechanics is able to describe the state of systems consisting of sub-atomic particles, such as neutrons, protons, or electrons. The need for the development of quantum mechanics came from the inability of classical physics to accurately describe some physical phenomena, as well as to predict their outcome. One of these phenomena is described in Figure 2.1, below.

$$\begin{tikzcd}
& & \boxed{O_2} & \\
& \boxed{M_2} \arrow[r] & \boxed{S_2} \arrow[r] \arrow[u] & \boxed{O_1} \\
\boxed{L} \arrow[r] & \boxed{S_1} \arrow[r] \arrow[u] & \boxed{M_1} \arrow[u] &
\end{tikzcd}$$

Figure 2.1: The setting of the experiment that its outcome is not well-described by classical physics. The symbol L denotes a light source. The symbols S_1 and S_2 denote beam splitters. The symbols M_1 and M_2 denote mirrors, and the symbols O_1 and O_2 denote observers.

What we see in Figure 2.1 is a light source L that is split in S_1, and then there emerge two light paths, namely

$$(L, S_1, M_2, S_2, O_1) = \text{path } 1,$$

and

$$(S_1, M_1, S_2, O_2) = \text{path } 2.$$

Now, we pose the question *what are the probabilities that the observers O_1 and O_2 observe photons?* According to the framework of classical physics, one has that each of the two observers, namely O_1 and O_2, receives light with probability 50%. However, whenever this experiment is performed, we get that

$$\Pr\left[\text{The light follows the path 1.}\right] = 0, \tag{2.32}$$

and

$$\Pr\left[\text{The light follows the path 2.}\right] = 1. \tag{2.33}$$

Yet, this experiment has a simple quantum-mechanical explanation. Let

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \tag{2.34}$$

denote the possibility the light follows the path 1, and let

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{2.35}$$

denote the possibility the light follows the path 2. If we perceive the splitter S_1 as a unitary

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & j \\ j & 1 \end{pmatrix}, \tag{2.36}$$

with $j^2 = -1$, then, after the light passes through it, we are at the state

$$\begin{aligned} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & j \\ j & 1 \end{pmatrix} |0\rangle &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & j \\ j & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ j \end{pmatrix}, \end{aligned} \tag{2.37}$$

and, after the light passes through S_2, which has the same unitary description as S_1, we are at the state

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & j \\ j & 1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ j \end{pmatrix} = \begin{pmatrix} 0 \\ j \end{pmatrix}. \tag{2.38}$$

Note that the mirrors M_1 and M_2 do not affect the state of our quantum system, as they only change the direction of the photon beam. Now, we ask the question: what does the state

$$\begin{aligned} \begin{pmatrix} 0 \\ j \end{pmatrix} &= 0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + j \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= 0 \cdot |0\rangle + j \cdot |1\rangle \\ &= \alpha_0 \cdot |0\rangle + \alpha_1 \cdot |1\rangle \end{aligned} \tag{2.39}$$

imply? It implies that

$$\Pr\left[\text{The light follows the path 1.}\right] = |\alpha_0|^2$$

$$\begin{aligned} &= |0|^2 \\ &= 0, \end{aligned} \tag{2.40}$$

and that

$$\begin{aligned} \Pr\left[\text{The light follows the path 2.}\right] &= |\alpha_1|^2 \\ &= |j|^2 \\ &= 1, \end{aligned} \tag{2.41}$$

for reasons that are a little blurry right now, see the Subsection 2.3.4. Thus, our quantum modeling seems to be very good for predictions! This very nice example of the power of quantum mechanics was drawn from the wonderful textbook by Kaye, Laflamme, and Mosca [83].

From a mathematical point of view, quantum mechanics can be perceived as a generalization of probability theory, see Table 2.1. That is, quantum mechanics is centered around the preservation of the L_2-norm, instead of the L_1-norm.

Table 2.1: Quantum mechanics can be viewed as a generalization of probability theory [8].

Probability Theory	Quantum Mechanics
Real numbers in $[0, 1]$	Complex numbers
Real numbers that sum to 1	Complex numbers that the squares of their magnitudes sum to 1
The *sum* is equal to 1	The *Euclidean norm* is equal to 1
The *sum* is preserved	The *Euclidean norm* is preserved
The L_1-norm is preserved	The L_2-norm is preserved
Use of stochastic matrices	Use of unitary matrices

2.3.1 Pure States

Pure quantum states are divided into one-qubit and many-qubit quantum states.

Pure One-Qubit States

A fundamental notion in quantum mechanics is that of the *qubit*. A *qubit*, or a quantum-bit, is a two-state system that is different from the ordinary classical bit in the sense that the qubit can be in a *quantum superposition* of some states $|0\rangle$ and $|1\rangle$, in contrast to the ordinary bit that either assumes the value 0 or the value 1. That is, for complex numbers $\alpha_1, \alpha_2 \in \mathbb{C}$, a qubit can be found in states of the form

$$|\psi\rangle = \alpha_1 |0\rangle + \alpha_2 |1\rangle \,, \tag{2.42}$$

subject to the constraint

$$\begin{aligned}\sum_{i=1}^{2} |\alpha_i|^2 &= |\alpha_1|^2 + |\alpha_2|^2 \\ &= \alpha_1^* \alpha_1 + \alpha_2^* \alpha_2 \\ &= 1.\end{aligned} \tag{2.43}$$

The quantity $|\alpha_1|^2$ denotes the probability that the qubit is found in state $|1\rangle$, after a measurement, and the quantity $|\alpha_0|^2$ denotes the probability that the qubit is found in the state $|0\rangle$, after a measurement.

Remark 2.18. *How many real numbers do we need to communicate the state of a qubit?* Well, at first this number seems to be four: two reals for each of the complexes α_1 and α_2. However, due to the constraint of (2.43), and to the fact that, for some reason not referred to, right now, we ignore overall phase factors, we need only two real numbers. That is, if we write each of the α_1 and α_2 in their polar form, that is,

$$\alpha_1 = r_1 \, e^{\phi_1 j}, \tag{2.44}$$

and

$$\alpha_2 = r_2 \, e^{\phi_2 j}, \tag{2.45}$$

we can see that

$$\begin{aligned}\alpha_1 |0\rangle + \alpha_2 |1\rangle &= r_1 \, e^{\phi_1 j} + r_2 \, e^{\phi_2 j} \\ &= e^{\phi_1 j} \left(r_1 + r_2 \, e^{(\phi_2 - \phi_1) j} \right).\end{aligned} \tag{2.46}$$

Now, if we take into consideration the fact that we do not account for overall phase factors, something that we will elaborate later on, in the Remark 2.46, we get that

$$\alpha_1 |0\rangle + \alpha_2 |1\rangle = r_1 + r_2 \, e^{(\phi_2 - \phi_1) j}. \tag{2.47}$$

Thus, if someone gives us the values of r_1 and $(\phi_2 - \phi_1)$ we have everything we need to fully understand the state (2.47) of our qubit. *Why?* Because we can use the equation

$$|\alpha_1|^2 + |\alpha_2|^2 = r_1^2 + r_2^2$$

$$= 1, \tag{2.48}$$

and the fact that $r_2 \geq 0$, to find the value of r_2. Thus, we now have all the information, that is necessary, to fully understand (2.47).

We define a particularly-simple orthonormal basis, for the representation of qubits, in the Definition 2.19.

Definition 2.19 (The Computational Basis)**.** We define the *computational basis* to be the set $\{|0\rangle, |1\rangle\}$, which consists of the kets

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \tag{2.49}$$

and

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{2.50}$$

Note that an arbitrary ket,

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \in \mathbb{C}^2, \tag{2.51}$$

can be decomposed as

$$|\psi\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{2.52}$$

Thus, the set $\{|0\rangle, |1\rangle\}$ is a basis, indeed, for one-qubit systems.

Remark 2.20 (Orthonormality of the Computational Basis)**.** Note that

$$\begin{aligned} \langle 0| &= (|0\rangle)^\dagger \\ &= \begin{pmatrix} 1 \\ 0 \end{pmatrix}^\dagger \\ &= \begin{pmatrix} 1 & 0 \end{pmatrix}, \end{aligned} \tag{2.53}$$

and that

$$\langle 1| = (|1\rangle)^\dagger$$

$$\begin{aligned} &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 1 \end{pmatrix}. \end{aligned} \tag{2.54}$$

Thus,

$$\begin{aligned} \langle 0|0\rangle &= \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= 1 \\ &= \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= \langle 1|1\rangle, \end{aligned} \tag{2.55}$$

and

$$\begin{aligned} \langle 0|1\rangle &= \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= 0 \\ &= \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \langle 1|0\rangle. \end{aligned} \tag{2.56}$$

These observations imply that the computational basis $\{|0\rangle, |1\rangle\}$ is orthogonal and normal, that is, it is orthonormal. Equivalently, for any two vectors, or kets, $|i\rangle, |j\rangle \in \{|0\rangle, |1\rangle\}$, one has that

$$\langle i|j\rangle = \delta_{i,j}. \tag{2.57}$$

This brings us to the Postulate 2.21.

Postulate 2.21 (State Space Postulate [83])**.** The state of a quantum system can be described by a unit vector in a complex vector space $\mathcal{H}$.

Postulate 2.21 is illustrated by the Example 2.22.

Example 2.22. Suppose that we have a one-qubit quantum system, in a state described by the ket $|\psi\rangle$. Let

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle$$

$$
\begin{aligned}
&= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{i}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\
&= \left(\frac{1}{\sqrt{2}}, \frac{i}{\sqrt{2}} \right) \in \mathbb{C}^2, \qquad (2.58)
\end{aligned}
$$

be that ket, expressed as a linear combination of the elements of the computational basis $\{|0\rangle, |1\rangle\}$. Observe that

$$
\begin{aligned}
\| |\psi\rangle \|_2 &= \sqrt{\left| \frac{1}{\sqrt{2}} \right|^2 + \left| \frac{i}{\sqrt{2}} \right|^2} \\
&= \sqrt{\left(\frac{1}{\sqrt{2}} \right)^2 + \left(\frac{1}{\sqrt{2}} \right)^2} \\
&= 1, \qquad (2.59)
\end{aligned}
$$

which translates to "$|\psi\rangle$ is a unit vector."

Pure Many-Qubit States

By combining together many one-qubit systems, that, of course, their corresponding state vectors are in $\mathbb{C}^2$, we can create bigger, many-qubit, systems. The way we denote the bigger space $\mathcal{H}_{\text{big}}$, of such a bigger many-qubit system, is the tensor product

$$
\begin{aligned}
\mathcal{H}_{\text{big}} &= \bigotimes_{i=1}^{n} (\mathcal{H}_{\text{small}}) \\
&= \bigotimes_{i=1}^{n} (\mathcal{H}_{\text{one qubit}}) \\
&= \bigotimes_{i=1}^{n} (\mathbb{C}^2) \\
&= \underbrace{\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2}_{n} \\
&= \left(\mathbb{C}^2 \right)^{\otimes n} \\
&= \mathbb{C}^{2^n}, \qquad (2.60)
\end{aligned}
$$

for the case that this bigger system is a system on n qubits. This brings us to Postulate 2.23.

Postulate 2.23 (Composite Systems Postulate [83]). When two quantum systems, namely S_1 and S_2, with vector spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, respectively, are treated as one combined system, the state space of the combined system is the tensor product $\mathcal{H}_1 \otimes \mathcal{H}_2$ of the vector spaces of the component systems. If the first system is in the state $|\psi_1\rangle \in \mathcal{H}_1$ and the second system is in the state $|\psi_2\rangle \in \mathcal{H}_2$, then the composite system is in the state

$$\begin{aligned} |\psi_1\rangle \otimes |\psi_2\rangle &= |\psi_1\rangle\, |\psi_2\rangle \\ &= |\psi_1\psi_2\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2. \end{aligned} \tag{2.61}$$

Remark 2.24. We now can derive (2.60), by repeatedly applying the underlying mechanism of the Postulate 2.23. That is, for

$$\begin{aligned} \mathcal{H}_1 &= \text{the space of a one-qubit system} \\ &= \mathbb{C}^2, \end{aligned} \tag{2.62}$$

and

$$\begin{aligned} \mathcal{H}_2 &= \text{the space of a one-qubit system} \\ &= \mathbb{C}^2, \end{aligned} \tag{2.63}$$

we get

$$\begin{aligned} \mathcal{H}_1 \otimes \mathcal{H}_2 &= \mathbb{C}^2 \otimes \mathbb{C}^2 \\ &= \left(\mathbb{C}^2\right)^{\otimes 2} \\ &= \mathbb{C}^{2^2} \\ &= \mathbb{C}^4 \\ &= \mathcal{H}_3. \end{aligned} \tag{2.64}$$

Thus, we created a two-qubit system. So, if we repeat this procedure for

$$\mathcal{H}_1 = \mathcal{H}_3, \tag{2.65}$$

and

$$\begin{aligned} \mathcal{H}_2 &= \text{the space of a one-qubit system} \\ &= \mathbb{C}^2, \end{aligned} \tag{2.66}$$

we are able to build a three-qubit system, with underlying space

$$\begin{aligned} \mathcal{H}_4 &= \mathbb{C}^{2^3} \\ &= \mathbb{C}^8, \end{aligned} \tag{2.67}$$

and so on, until we reach the desired n-qubit quantum system.

Definition 2.25 (The Computational Basis, Revisited)**.** The *computational basis*, for describing the basis state vectors, of quantum systems on n qubits, is defined to be

$$B = \{|y\rangle\}_{y\in\{0,1\}^n} . \tag{2.68}$$

That is, the computational basis encodes all of the possible strings, of length equal to n, whose elements are drawn from the set $\{0,1\}$. These elements, correspond to all of the possible configurations of our n-qubit quantum computer. We now set

$$|y\rangle = \left.\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix}\right\} 2^n, \tag{2.69}$$

where the "1" is in the y-th position, after we convert $y \in B$ from binary to decimal.

Note 2.26 (Orthonormality of the Computational Basis, Revisited)**.** As one observes, the computational basis, presented in the Definition 2.25, is orthonormal.

Example 2.27. Suppose that we have some system S_1 in the state

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle , \tag{2.70}$$

and some system S_2 in the state

$$|\psi_2\rangle = |0\rangle . \tag{2.71}$$

Then, the combined system is in the state

$$\begin{aligned} |\psi_1\rangle \otimes |\psi_2\rangle &= \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes |0\rangle \\ &= \frac{1}{\sqrt{2}}|0\rangle \otimes |0\rangle + \frac{1}{\sqrt{2}}|1\rangle \otimes |0\rangle \\ &= \frac{1}{\sqrt{2}}|0\rangle|0\rangle + \frac{1}{\sqrt{2}}|1\rangle|0\rangle \\ &= \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle . \end{aligned} \tag{2.72}$$

Note that

$$\begin{aligned} \| |\psi_1\rangle \otimes |\psi_2\rangle \|_2 &= \sqrt{\left|\frac{1}{\sqrt{2}}\right|^2 + \left|\frac{1}{\sqrt{2}}\right|^2} \\ &= 1, \end{aligned} \tag{2.73}$$

which means that the new composite vector is a unit-vector, and, thus, it satisfies the Postulate 2.21.

Note 2.28 (Alternative Basis-Representation)**.** Many times, we denote the computational basis

$$B = \{|y\rangle\}_{y\in\{0,1\}^n}, \tag{2.74}$$

as

$$D = \{|i\rangle\}_{i\in[2^n]}, \tag{2.75}$$

by taking advantage of the fact that there is a one-to-one, and onto, function between the elements of B and D, or, equivalently,

$$|B| = |D| = 2^n. \tag{2.76}$$

That is, every string in B is matched to a unique number in D, and vice-versa. This correspondence is the representation of the binary "numbers," of B, as decimal numbers of D, and vice-versa. The elements of D are often called "qudits," that is, "quantum digits."

Note 2.29. Suppose that we have a quantum system on n qubits. Then, its state, say $|\psi\rangle$, may be written uniquely as a combination of the elements of the computational basis, as

$$\begin{aligned} |\psi\rangle &= \sum_{j\in B} \alpha_j |j\rangle \\ &= \sum_{j\in\{0,1\}^n} \alpha_j |j\rangle, \end{aligned} \tag{2.77}$$

or, equivalently,

$$\begin{aligned} |\psi\rangle &= \sum_{i\in D} \alpha_i |i\rangle \\ &= \sum_{i=1}^{2^n} \alpha_i |i\rangle. \end{aligned} \tag{2.78}$$

Remark 2.30 (Inner-Products)**.** Suppose that we are given two vectors, or kets, that live in $\mathbb{C}^{2^n}$, for some $n \in \mathbb{N}$, namely

$$|\psi\rangle = \sum_{i=1}^{2^n} \alpha_i |i\rangle \tag{2.79}$$

and

$$|\phi\rangle = \sum_{i=1}^{2^n} \beta_i |i\rangle . \tag{2.80}$$

Note that

$$\begin{aligned} \langle\psi| &= (|\psi\rangle)^\dagger \\ &= \sum_{i=1}^{2^n} (\alpha_i |i\rangle)^\dagger \\ &= \sum_{i=1}^{2^n} \alpha_i^\dagger \langle i| \\ &= \sum_{i=1}^{2^n} \alpha_i^* \langle i| . \end{aligned} \tag{2.81}$$

The *inner-product*, of $|\psi\rangle$ and $|\phi\rangle$, is a function

$$\mathcal{I} : \mathcal{H} \times \mathcal{H} \to \mathbb{C}, \tag{2.82}$$

such that

$$\begin{aligned} \mathcal{I}(|\psi\rangle, |\phi\rangle) &= |\psi\rangle^\dagger \cdot |\phi\rangle \\ &= \langle\psi| \, |\phi\rangle \\ &= \langle\psi|\phi\rangle \\ &= \left(\sum_{i=1}^{2^n} \alpha_i^* \langle i|\right) \left(\sum_{i=1}^{2^n} \beta_i |i\rangle\right) \\ &= \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} \alpha_i^* \beta_j \langle i|j\rangle \\ &= \sum_{i=1}^{2^n} \sum_{j=1}^{2^n} \alpha_i^* \beta_j \delta_{i,j} \\ &= \sum_{i=1}^{2^n} \alpha_i^* \beta_i. \end{aligned} \tag{2.83}$$

Note that the second part of (2.83) is a number.

Example 2.31. Suppose that we are given the vectors

$$|\psi\rangle = \begin{pmatrix} a \\ b \end{pmatrix} \in \mathbb{C}^2, \tag{2.84}$$

and

$$|\phi\rangle = \begin{pmatrix} c \\ d \end{pmatrix} \in \mathbb{C}^2. \tag{2.85}$$

Then, one has that

$$\begin{aligned} \mathcal{I}\left(|\psi\rangle, |\phi\rangle\right) &= |\psi\rangle^\dagger \cdot |\phi\rangle \\ &= \langle\psi|\,|\phi\rangle \\ &= \langle\psi|\phi\rangle \\ &= \left(a^* \quad b^*\right) \begin{pmatrix} c \\ d \end{pmatrix} \\ &= a^*c + b^*d. \end{aligned} \tag{2.86}$$

Note 2.32. We have that

$$\begin{aligned} \langle\psi|\phi\rangle &= \left(\langle\phi|\psi\rangle\right)^* \\ &= \langle\phi|\psi\rangle^*. \end{aligned} \tag{2.87}$$

That is, by changing the order in an inner-product, we get the complex-conjugate of the initial inner-product value. Alternatively, inner-products constitute ordered operations.

Remark 2.33 (L_2-Norms, Revisited)**.** Note, for some ket $|\psi\rangle$, that

$$\begin{aligned} \||\psi\rangle\|_2 &= \sqrt{\mathcal{I}\left(|\psi\rangle, |\psi\rangle\right)} \\ &= \sqrt{|\psi\rangle^\dagger \cdot |\psi\rangle} \\ &= \sqrt{\langle\psi|\psi\rangle}, \end{aligned} \tag{2.88}$$

or

$$\||\psi\rangle\|_2^2 = \langle\psi|\psi\rangle. \tag{2.89}$$

Remark 2.34 (Outer-Products)**.** Suppose that we are given two vectors, or kets, $|\psi\rangle$ and $|\phi\rangle$, as in the Remark 2.30. The *outer-product* of the vectors $|\psi\rangle$ and $|\phi\rangle$, that live in the vector space $\mathcal{H}$, and have dimension 2^n, for some natural n, is a function

$$\mathcal{O} : \mathcal{H} \times \mathcal{H} \to \mathbb{C}^{2^n \times 2^n}, \tag{2.90}$$

with

$$\begin{aligned} \mathcal{O}\left(|\psi\rangle, |\phi\rangle\right) &= |\psi\rangle\langle\phi| \\ &= |\psi\rangle \cdot |\phi\rangle^{\dagger} \\ &= \mathcal{I}\left(|\psi\rangle^{\dagger}, |\phi\rangle^{\dagger}\right) \\ &= \mathcal{I}\left(|\psi\rangle^{\dagger}, \langle\phi|\right) \\ &= \left(\sum_{i=1}^{2^n} \alpha_i\, |i\rangle\right)\left(\sum_{i=1}^{2^n} \beta_i^*\, \langle i|\right) \\ &= \sum_{i=1}^{2^n}\sum_{j=1}^{2^n} \alpha_i \beta_j^*\, |i\rangle\langle j|\,. \end{aligned} \tag{2.91}$$

Note that the second part of (2.91) is a $2^n \times 2^n$ matrix.

Example 2.35. Suppose that we are given the vectors

$$|\psi\rangle = \begin{pmatrix} a \\ b \end{pmatrix} \in \mathbb{C}^2, \tag{2.92}$$

and

$$|\phi\rangle = \begin{pmatrix} c \\ d \end{pmatrix} \in \mathbb{C}^2. \tag{2.93}$$

Then, one has that

$$\begin{aligned} \mathcal{O}\left(|\psi\rangle, |\phi\rangle\right) &= |\psi\rangle\langle\phi| \\ &= |\psi\rangle\, |\phi\rangle^{\dagger} \\ &= \begin{pmatrix} a \\ b \end{pmatrix}\begin{pmatrix} c \\ d \end{pmatrix}^{\dagger} \\ &= \begin{pmatrix} a \\ b \end{pmatrix}\begin{pmatrix} c^* & d^* \end{pmatrix} \end{aligned}$$

$$\begin{aligned}
&= \begin{pmatrix} ac^* & ad^* \\ bc^* & bd^* \end{pmatrix} \\
&= ac^* \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + ad^* \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \\
&\quad + bc^* \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} + bd^* \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \\
&= ac^* \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix} + ad^* \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \end{pmatrix} \\
&\quad + bc^* \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix} + bd^* \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \end{pmatrix} \\
&= ac^* |0\rangle\langle 0| + ad^* |0\rangle\langle 1| + bc^* |1\rangle\langle 0| + bd^* |1\rangle\langle 1| . \qquad (2.94)
\end{aligned}$$

2.3.2 Mixed States

There are cases at which our quantum system is not described by a pure state, but rather a mixed one.

Definition 2.36 (Mixed States)**.** Let $k \in \mathbb{N}$. A *mixed state* is a probability distribution on some pure states, that is, it is the set of 2-tuples

$$\{(p_i, |\psi_i\rangle)\}_{i\in[k]} = \{(p_1, |\psi_1\rangle), (p_2, |\psi_2\rangle), \dots, (p_k, |\psi_k\rangle)\}, \qquad (2.95)$$

with

$$\sum_{i=1}^{k} p_i = 1. \qquad (2.96)$$

Remark 2.37 (Pure States as Mixed States)**.** Every pure state is a mixed one. That is, a pure state $|\psi\rangle$ can be sought as a distribution

$$\begin{aligned}
\{(p_i, |\psi_i\rangle)\}_{i\in[k]} &= \{(p_1, |\psi_1\rangle), (p_2, |\psi_2\rangle), \dots, (p_k, |\psi_k\rangle)\} \\
&= \{(1, |\psi\rangle), (0, |\psi\rangle), \dots, (0, |\psi\rangle)\} . \qquad (2.97)
\end{aligned}$$

We treat mixed states by using density matrices, see Definition 2.38.

Definition 2.38 (Density Matrices)**.** Let $k \in \mathbb{N}$. A mixed state

$$\{(p_i, |\psi_i\rangle)\}_{i\in[k]} = \{(p_1, |\psi_1\rangle), (p_2, |\psi_2\rangle), \ldots, (p_k, |\psi_k\rangle)\}, \tag{2.98}$$

is described by a *density matrix*, namely ρ, that encodes all of the information a mixed state carries. We have that

$$\rho = \sum_{i=1}^{k} p_i |\psi_i\rangle\langle\psi_i|. \tag{2.99}$$

Remark 2.39 (Density Matrices for Pure States)**.** The density matrix of some pure state $|\psi\rangle$ is

$$\begin{aligned}\rho &= \sum_{i=1}^{k} p_i |\psi_i\rangle\langle\psi_i| \\ &= \underbrace{1 \cdot |\psi\rangle\langle\psi| + 0 \cdot |\psi\rangle\langle\psi| + \cdots + 0 \cdot |\psi\rangle\langle\psi|}_{k} \\ &= |\psi\rangle\langle\psi|. \end{aligned} \tag{2.100}$$

2.3.3 Time-Evolution

As it turns out, quantum systems evolve in a unitary fashion. Unitary evolution preserves the L_2-norm.

Time-Evolution of Pure States

Postulate 2.40 (Time-Evolution Postulate [83])**.** The time-evolution of a closed quantum system $\mathcal{Q}$ can be described by a unitary transformation over the Hilbert space that holds the unit-vectors, that is, the kets, that describe its state. That is, if the ket $|\psi_1\rangle \in \mathcal{H}$ denotes a state of a quantum system $\mathcal{Q}$, and the ket $|\psi_2\rangle \in \mathcal{H}$ denotes some later, in time, state, of the same quantum system $\mathcal{Q}$, then there is some unitary transformation U, such that

$$U|\psi_1\rangle = |\psi_2\rangle, \tag{2.101}$$

and

$$\langle\psi_1| U^\dagger = (U|\psi_1\rangle)^\dagger$$

$$\begin{aligned} &= |\psi_2\rangle^\dagger \\ &= \langle\psi_2| \,. \end{aligned} \tag{2.102}$$

Postulate 2.40 is illustrated by the Example 2.41.

Example 2.41. Let

$$\begin{aligned} |\psi_1\rangle &= |0\rangle \\ &= \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \end{aligned} \tag{2.103}$$

be the initial state, and

$$\begin{aligned} |\psi_2\rangle &= \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \\ &= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \frac{1}{\sqrt{2}}\begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \end{aligned} \tag{2.104}$$

be the final state. There is a unitary tranformation U which governs the evolution of this one-qubit quantum system, or, in math terms,

$$U|\psi_1\rangle = |\psi_2\rangle \,. \tag{2.105}$$

Here, the unitary U is such that

$$U = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{2.106}$$

Note that U is, indeed, unitary, since if we take into consideration the fact that

$$\begin{aligned} U^\dagger &= \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\ &= U^T, \end{aligned} \tag{2.107}$$

we get that

$$UU^\dagger = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$
\begin{aligned}
&= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\
&= I_2, \qquad (2.108)
\end{aligned}
$$

and

$$
\begin{aligned}
U^\dagger U &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\
&= I_2. \qquad (2.109)
\end{aligned}
$$

In other words, we have $U^{-1} = U^\dagger$. Therefore U is unitary.

Time-Evolution of Mixed States

Remark 2.42 (Time-Evolution of a Mixed State [83]). As we saw earlier, pure states evolve unitarily. So, one observes that

$$
\{(p_i, |\psi_i\rangle)\}_{i\in[k]} = \{(p_1, |\psi_1\rangle), (p_2, |\psi_2\rangle), \ldots, (p_k, |\psi_k\rangle)\}, \qquad (2.110)
$$

and, after the unitary transformation U is applied, we get

$$
\{(p_i, U|\psi_i\rangle)\}_{i\in[k]} = \{(p_1, U|\psi_1\rangle), (p_2, U|\psi_2\rangle), \ldots, (p_k, U|\psi_k\rangle)\}, \qquad (2.111)
$$

and, thus,

$$
\begin{aligned}
\rho_{\text{new}} &= \sum_{i=1}^{k} p_i \left(U|\psi_i\rangle\right)\left(\langle\psi_i| U^\dagger\right) \\
&= \sum_{i=1}^{k} p_i U |\psi_i\rangle\langle\psi_i| U^\dagger \\
&= U\left(\sum_{i=1}^{k} p_i |\psi_i\rangle\langle\psi_i|\right) U^\dagger \\
&= U\rho U^\dagger. \qquad (2.112)
\end{aligned}
$$

Definition 2.43 (CPTP maps [83]). There is yet another way of a mixed state to evolve: a somewhat more generic way. A *completely-positive trace-preserving* map, or a CPTP map, is a function that maps density matrices to density matrices, or

$$\rho \mapsto \sum_i A_i \rho A_i^\dagger, \tag{2.113}$$

with

$$\sum_i A_i^\dagger A_i = I. \tag{2.114}$$

The set $\{A_i\}$ holds the *Kraus operators*, which are linear operators satisfying the Equation (2.114).

2.3.4 Measurements

In order to access the information that lives in some quantum system, we need to measure the respective system. Measurements are total or partial, and refer to pure or mixed states.

We begin by introducing measurements for pure quantum states, be them on one or many qubits. First, we introduce the concept of projectors.

Definition 2.44 (Projectors)**.** A *projector*, on a vector space $\mathcal{H}$, is a linear transformation P, such that

$$P^2 = P, \tag{2.115}$$

and, if it is an *orthogonal projector*,

$$P^\dagger = P. \tag{2.116}$$

Using these projectors, one is able to define "projective measurements." See Postulate 2.45.

Postulate 2.45 (Measurement Postulate [83])**.** Let $N \in \mathbb{N}$. For a given orthonormal vector basis

$$B = \{\varphi_i\}_{i \in [N]}, \tag{2.117}$$

we can write any quantum state, while $\alpha_i \in \mathbb{C}$, for every $i \in [N]$, as

$$|\psi\rangle = \sum_{i=1}^{N} \alpha_i |\varphi_i\rangle,$$

with

$$\begin{aligned} \| |\psi\rangle \|_2^2 &= \sum_{i=1}^{N} |\alpha_i|^2 \\ &= 1. \end{aligned} \tag{2.118}$$

Let $\{P_i\}_{i\in[N]}$ be a set of N orthogonal projectors, that is, one for each of the N subspaces that, in turn, correspond, one to one, to the N possible measurement outcomes. Note that, here,

$$\sum_{i=1}^{N} P_i = I. \tag{2.119}$$

After we perform a measurement, and if we let M denote the measurement outcome, we have that

$$\begin{aligned} \Pr\left[\text{The outcome is ``}i.\text{''}\right] &= \Pr\left[M = i\right] \\ &= \|P_i\,|\psi\rangle\|_2^2 \\ &= \left(\sqrt{\mathcal{I}\left(P_i\,|\psi\rangle\,, P_i\,|\psi\rangle\right)}\right)^2 \\ &= \mathcal{I}\left(P_i\,|\psi\rangle\,, P_i\,|\psi\rangle\right) \\ &= \left(P_i\,|\psi\rangle\right)^\dagger \left(P_i\,|\psi\rangle\right) \\ &= |\psi\rangle^\dagger\, P_i^\dagger P_i\,|\psi\rangle \\ &= \langle\psi|P_i P_i|\psi\rangle \\ &= \langle\psi|P_i^2|\psi\rangle \\ &= \langle\psi|P_i|\psi\rangle, \end{aligned} \tag{2.120}$$

or

$$\begin{aligned} \Pr\left[\text{The outcome is ``}i.\text{''}\right] &= \langle\psi|P_i|\psi\rangle \\ &= \mathrm{Tr}\left(\langle\psi|P_i|\psi\rangle\right) \\ &= \mathrm{Tr}\left(P_i\,|\psi\rangle\langle\psi|\right) \\ &= \mathrm{Tr}\left(P_i\,\rho\right), \end{aligned} \tag{2.121}$$

for ρ being the density matrix that corresponds to the pure state $|\psi\rangle$. After a measurement, the state of the quantum system collapses to the state observed in the measurement. In particular, it collapses to the state

$$|\phi_i\rangle = \frac{P_i\,|\psi\rangle}{\|P_i\,|\psi\rangle\|_2}$$

$$\begin{aligned} &= \frac{P_i \left|\psi\right\rangle}{\sqrt{\langle\psi|P_i|\psi\rangle}} \\ &= \frac{P_i \left|\psi\right\rangle}{\sqrt{\mathrm{Tr}\left(P_i\,\rho\right)}}. \end{aligned} \tag{2.122}$$

Note that

$$\begin{aligned} \|\left|\phi_i\right\rangle\|_2^2 &= \frac{\|P_i \left|\psi\right\rangle\|_2^2}{\mathrm{Tr}\left(P_i\,\rho\right)} \\ &= \frac{\mathrm{Tr}\left(P_i\,\rho\right)}{\mathrm{Tr}\left(P_i\,\rho\right)} \\ &= 1, \end{aligned} \tag{2.123}$$

which means that $\left|\phi_i\right\rangle$ is a unit-vector.

Remark 2.46 (Neglecting Overall Phase Factors)**.** *Why do we neglect, in our calculations, overall phase factors?* The reason is that any phase factor does change the outcome of any measurement made. Thus, since their effects are not subject to identification, via measurements, we ignore them completely. To be more precise, suppose that we have an n-qubit state

$$\left|\theta\right\rangle = \sum_{i=1}^{2^n} \alpha_i \left|i\right\rangle, \tag{2.124}$$

and an n-qubit state, for $\varphi \in \mathbb{R}$, and $j^2 = -1$,

$$\begin{aligned} \left|\phi\right\rangle &= e^{j\varphi} \left|\theta\right\rangle \\ &= e^{j\varphi} \sum_{i=1}^{2^n} \alpha_i \left|i\right\rangle \\ &= \sum_{i=1}^{2^n} e^{j\varphi} \alpha_i \left|i\right\rangle. \end{aligned} \tag{2.125}$$

Note now that

$$\begin{aligned} \langle\phi|P_i|\phi\rangle &= \left(\left\langle\theta\right| \left(e^{j\varphi}\right)^*\right) P_i \left(e^{j\varphi}\left|\theta\right\rangle\right) \\ &= \left(\left\langle\theta\right| e^{-j\varphi}\right) P_i \left(e^{j\varphi}\left|\theta\right\rangle\right) \\ &= \langle\theta|P_i|\theta\rangle e^{-j\varphi} e^{j\varphi} \\ &= \langle\theta|P_i|\theta\rangle. \end{aligned} \tag{2.126}$$

That is,

Probability to measure "i," in state $\left|\theta\right\rangle$

$$\begin{aligned}&= \|P_i \left|\theta\right\rangle\|_2^2 \\ &= \langle\theta|P_i|\theta\rangle \\ &= \langle\phi|P_i|\phi\rangle \\ &= \|P_i \left|\phi\right\rangle\|_2^2 \\ &= \text{Probability to measure “}i\text{,” in state } \left|\phi\right\rangle. \qquad (2.127)\end{aligned}$$

Remark 2.47 (Von Neumann Measurements)**.** In the case that the projectors are of rank one, then we have a Von Neumann measurement. A projector P is of rank one, if there is some pure state, namely $\left|\psi\right\rangle$, with

$$\begin{aligned}\|\left|\psi\right\rangle\|_2^2 &= \left(\sqrt{\mathcal{I}\left(\left|\psi\right\rangle, \left|\psi\right\rangle\right)}\right)^2 \\ &= \mathcal{I}\left(\left|\psi\right\rangle, \left|\psi\right\rangle\right) \\ &= \left|\psi\right\rangle^\dagger \cdot \left|\psi\right\rangle \\ &= \left\langle\psi\right| \cdot \left|\psi\right\rangle \\ &= \langle\psi|\psi\rangle \\ &= 1, \qquad (2.128)\end{aligned}$$

such that

$$P = \left|\psi\right\rangle\left\langle\psi\right|. \qquad (2.129)$$

Observe that

$$\begin{aligned}P^\dagger &= \left(\left|\psi\right\rangle\left\langle\psi\right|\right)^\dagger \\ &= \left\langle\psi\right|^\dagger \left|\psi\right\rangle^\dagger \\ &= \left|\psi\right\rangle\left\langle\psi\right| \\ &= P, \qquad (2.130)\end{aligned}$$

and that

$$\begin{aligned}P^2 &= \left|\psi\right\rangle\left\langle\psi\right| \left|\psi\right\rangle\left\langle\psi\right| \\ &= \left|\psi\right\rangle \langle\psi|\psi\rangle \left\langle\psi\right| \\ &= \left|\psi\right\rangle \cdot 1 \cdot \left\langle\psi\right| \\ &= \left|\psi\right\rangle\left\langle\psi\right| \\ &= P. \qquad (2.131)\end{aligned}$$

Example 2.48 (Von Neumann Measurements)**.** Let $\{|0\rangle, |1\rangle\}$ denote the computational basis, and let

$$\begin{aligned}|\psi\rangle &= \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \\ &= \alpha_0|0\rangle + \alpha_1|1\rangle \end{aligned} \tag{2.132}$$

be the state of a quantum system on one qubit. Let, also,

$$P_0 = |0\rangle\langle 0| \tag{2.133}$$

and

$$P_1 = |1\rangle\langle 1| \tag{2.134}$$

be the elements of the set

$$\{P_i\}_{i\in\{0,1\}} = \{P_0, P_1\} \tag{2.135}$$

that contains the measurement projectors, each for a possible measurement outcome. After we perform a Von Neumann measurement, we have that

$$\begin{aligned}\Pr[\text{The outcome is "0."}] &= \langle\psi|P_0|\psi\rangle \\ &= \langle\psi||0\rangle\langle 0||\psi\rangle \\ &= \langle\psi|0\rangle\langle 0|\psi\rangle \\ &= \langle\psi|0\rangle\langle\psi|0\rangle^* \\ &= |\langle\psi|0\rangle|^2 \\ &= \left|\left(\frac{1}{\sqrt{2}}\langle 0| + \frac{1}{\sqrt{2}}\langle 1|\right)|0\rangle\right|^2 \\ &= \left|\frac{1}{\sqrt{2}}\langle 0|0\rangle + \frac{1}{\sqrt{2}}\langle 1|0\rangle\right|^2 \\ &= \left|\frac{1}{\sqrt{2}}\cdot 1 + \frac{1}{\sqrt{2}}\cdot 0\right|^2 \\ &= \left|\frac{1}{\sqrt{2}}\right|^2 \\ &= |\alpha_0|^2 \\ &= \frac{1}{2}. \end{aligned} \tag{2.136}$$

After this measurement, the system is in the state

$$|\phi_0\rangle = \frac{P_0|\psi\rangle}{\sqrt{\langle\psi|P_0|\psi\rangle}}$$

$$
\begin{aligned}
&= \frac{|0\rangle\langle 0| \left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right)}{\frac{1}{\sqrt{2}}} \\
&= \frac{\frac{1}{\sqrt{2}} |0\rangle}{\frac{1}{\sqrt{2}}} \\
&= |0\rangle . \qquad (2.137)
\end{aligned}
$$

In a similar way, one can show that

$$
\Pr\left[\text{The outcome is “1.”}\right] = \frac{1}{2}, \qquad (2.138)
$$

while the system is left in the state

$$
\begin{aligned}
|\phi_1\rangle &= \frac{P_1 |\psi\rangle}{\sqrt{\langle \psi | P_1 | \psi \rangle}} \\
&= \frac{|1\rangle\langle 1| \left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right)}{\frac{1}{\sqrt{2}}} \\
&= \frac{\frac{1}{\sqrt{2}} |1\rangle}{\frac{1}{\sqrt{2}}} \\
&= |1\rangle . \qquad (2.139)
\end{aligned}
$$

Remark 2.49 (Measurements and Mixed States)**.** Suppose that, for $k \in \mathbb{N}$, we have a quantum system which is described by a mixed state

$$
\rho = \sum_{i=1}^{k} p_i |\psi_i\rangle . \qquad (2.140)
$$

We have, for M being the random variable that denotes the outcome of some measurement, and P_j being an orthogonal projector into the subspace corresponding to the “j” measurement outcome, that

$$
\begin{aligned}
\Pr\left[\text{The outcome is “}j\text{.”}\right] &= \Pr\left[M = j\right] \\
&= \sum_{i=1}^{k} p_i \left\| P_j |\psi_i\rangle \right\|_2^2 \\
&= \sum_{i=1}^{k} p_i \left(\sqrt{\left(P_j |\psi_i\rangle\right)^\dagger P_j |\psi_i\rangle} \right)^2 \\
&= \sum_{i=1}^{k} p_i \left(\sqrt{|\psi_i\rangle^\dagger P_j^\dagger P_j |\psi_i\rangle} \right)^2
\end{aligned}
$$

$$\begin{aligned}
&= \sum_{i=1}^{k} p_i \left(\sqrt{\langle \psi_i | P_j P_j | \psi_i \rangle} \right)^2 \\
&= \sum_{i=1}^{k} p_i \left(\sqrt{\langle \psi_i | P_j^2 | \psi_i \rangle} \right)^2 \\
&= \sum_{i=1}^{k} p_i \left(\sqrt{\langle \psi_i | P_j | \psi_i \rangle} \right)^2 \\
&= \sum_{i=1}^{k} p_i \langle \psi_i | P_j | \psi_i \rangle \\
&= \sum_{i=1}^{k} p_i \mathrm{Tr} \left(\langle \psi_i | P_j | \psi_i \rangle \right) \\
&= \sum_{i=1}^{k} p_i \mathrm{Tr} \left(P_j | \psi_i \rangle \langle \psi_i | \right) \\
&= \sum_{i=1}^{k} \mathrm{Tr} \left(p_i P_j | \psi_i \rangle \langle \psi_i | \right) \\
&= \mathrm{Tr} \left(\sum_{i=1}^{k} p_i P_j | \psi_i \rangle \langle \psi_i | \right) \\
&= \mathrm{Tr} \left(P_j \sum_{i=1}^{k} p_i | \psi_i \rangle \langle \psi_i | \right) \\
&= \mathrm{Tr} \left(P_j \rho \right) . \qquad (2.141)
\end{aligned}$$

So, we can compute, in this way, the associated outcome probabilities for mixed states. Now, what is the new mixed state, that our quantum system collapses to, after the measurement? It is the mixed state with density matrix

$$\begin{aligned}
\rho_{\text{after}} &= \frac{P_j \rho_{\text{before}} P_j^\dagger}{\mathrm{Tr} \left(P_j \rho_{\text{before}} \right)} \\
&= \frac{P_j \rho P_j^\dagger}{\mathrm{Tr} \left(P_j \rho \right)} \\
&= \frac{P_j \rho P_j}{\mathrm{Tr} \left(P_j \rho \right)} . \qquad (2.142)
\end{aligned}$$

Remark 2.50 (Observables)**.** An *observable* is an Hermitean matrix that is somehow related to measurements. In quantum mechanics, measurable quantities, like the position or the momentum of a particle, are represented with Hermitean matrices, called observables. That is, the eigenvalues $\{\lambda_i\}_i$ of the corresponding observable are the possible outcomes of the measurement, and its eigenvectors $\{|\lambda_i\rangle\}_i$ are used to calculate the respective outcome-probabilities. We have, by the spectral theorem, that

$$A = \sum_{i=1}^{k} \lambda_i \left|\lambda_i\right\rangle \left\langle\lambda_i\right| . \tag{2.143}$$

The probability that the outcome of the measurement is λ_i, for every i, is, if we let M denote the random variable that corresponds to the outcome of the measurement,

$$\begin{aligned}
\Pr\left[\text{The outcome is "}\lambda_i\text{."}\right] &= \Pr\left[M = \lambda_i\right] \\
&= \left\|\Pi_{\lambda_i} \left|\psi\right\rangle\right\|_2^2 \\
&= \left\|\left|\lambda_i\right\rangle\left\langle\lambda_i\right| \left|\psi\right\rangle\right\|_2^2 \\
&= \left(\sqrt{\left(\left(\left|\lambda_i\right\rangle\left\langle\lambda_i\right|\right)\left|\psi\right\rangle\right)^\dagger \left(\left|\lambda_i\right\rangle\left\langle\lambda_i\right|\right)\left|\psi\right\rangle}\right)^2 \\
&= \left(\sqrt{\left|\psi\right\rangle^\dagger \left(\left|\lambda_i\right\rangle\left\langle\lambda_i\right|\right)^\dagger \left(\left|\lambda_i\right\rangle\left\langle\lambda_i\right|\right)\left|\psi\right\rangle}\right)^2 \\
&= \left\langle\psi\right| \left(\left|\lambda_i\right\rangle\left\langle\lambda_i\right|\right)^\dagger \left(\left|\lambda_i\right\rangle\left\langle\lambda_i\right|\right)\left|\psi\right\rangle \\
&= \left\langle\psi\right| \left(\left|\lambda_i\right\rangle\left\langle\lambda_i\right|\right) \left(\left|\lambda_i\right\rangle\left\langle\lambda_i\right|\right)\left|\psi\right\rangle \\
&= \left\langle\psi\right| \left(\left|\lambda_i\right\rangle\left\langle\lambda_i\right|\right)^2 \left|\psi\right\rangle \\
&= \left\langle\psi\right| \left|\lambda_i\right\rangle\left\langle\lambda_i\right| \left|\psi\right\rangle \\
&= \left\langle\psi|\lambda_i\right\rangle \left\langle\lambda_i|\psi\right\rangle \\
&= \left\langle\psi|\lambda_i\right\rangle \left\langle\psi|\lambda_i\right\rangle^* \\
&= \left|\left\langle\psi|\lambda_i\right\rangle\right|^2 .
\end{aligned} \tag{2.144}$$

The expected value, that is, the expectation, of the observable A is equal to

$$\begin{aligned}
\left\langle A\right\rangle &= \sum_{i=1}^{k} \lambda_i \Pr\left[M = \lambda_i\right] \\
&= \sum_{i=1}^{k} \lambda_i \left|\left\langle\psi|\lambda_i\right\rangle\right|^2 .
\end{aligned} \tag{2.145}$$

Definition 2.51 (Pauli Matrices)**.** The Pauli matrices, namely σ_x, σ_y, and σ_z, are, when expressed in the computational basis,

$$\begin{aligned}
\sigma_x &= \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\
&= \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}
\end{aligned}$$

$$= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \end{pmatrix}$$
$$= |1\rangle\langle 0| + |0\rangle\langle 1|, \tag{2.146}$$

and

$$\sigma_y = \begin{pmatrix} 0 & -j \\ j & 0 \end{pmatrix}$$
$$= j\,|1\rangle\langle 0| - j\,|0\rangle\langle 1|, \tag{2.147}$$

and

$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$
$$= |0\rangle\langle 0| - |1\rangle\langle 1|, \tag{2.148}$$

with $j^2 = -1$.

Note 2.52. Note that there exist some alternative names for the Pauli matrices, namely

$$\sigma_x = \sigma_1 = X, \tag{2.149}$$
$$\sigma_y = \sigma_2 = Y, \tag{2.150}$$

and

$$\sigma_z = \sigma_3 = Z. \tag{2.151}$$

Example 2.53. We consider the observable

$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{2.152}$$

which is the "third" Pauli matrix. To find its eigenvalues, we, at first, form the matrix, for some λ,

$$\sigma_z - \lambda I = \begin{pmatrix} 1-\lambda & 0 \\ 0 & -1-\lambda \end{pmatrix} \tag{2.153}$$

and, then, we compute the determinant

$$\begin{aligned}\det\left(\sigma_z - \lambda I\right) &= \det\begin{pmatrix} 1-\lambda & 0 \\ 0 & -1-\lambda \end{pmatrix} \\ &= (1-\lambda)(-1-\lambda) \\ &= -\left(1-\lambda^2\right) \\ &= \lambda^2 - 1. \end{aligned} \tag{2.154}$$

We now have that

$$\begin{aligned} &\det\left(\sigma_z - \lambda I\right) = 0 \\ \Rightarrow\ &\lambda^2 - 1 = 0 \\ \Rightarrow\ &\lambda = 1, \text{ or } \lambda = -1. \end{aligned} \tag{2.155}$$

Thus, the eigenvalues of σ_z are $\lambda_1 = 1$ and $\lambda_2 = -1$. To find the eigenvectors of σ_z, we devise the equations

$$\sigma_z \left|\lambda_1\right\rangle = \lambda_1 \left|\lambda_1\right\rangle, \tag{2.156}$$

and

$$\sigma_z \left|\lambda_2\right\rangle = \lambda_2 \left|\lambda_2\right\rangle. \tag{2.157}$$

Let

$$\left|\lambda_1\right\rangle = \begin{pmatrix} a \\ b \end{pmatrix}. \tag{2.158}$$

So, we have that

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}, \tag{2.159}$$

or

$$\begin{pmatrix} a \\ -b \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}, \tag{2.160}$$

or

$$-b = b, \tag{2.161}$$

or

$$b = 0. \tag{2.162}$$

Thus, we now know that

$$|\lambda_1\rangle = \begin{pmatrix} a \\ 0 \end{pmatrix}. \tag{2.163}$$

To find a, we need to take into consideration the fact that

$$\langle\lambda_1|\lambda_1\rangle = 1, \tag{2.164}$$

or

$$\begin{pmatrix} a^* & 0 \end{pmatrix} \begin{pmatrix} a \\ 0 \end{pmatrix} = 1, \tag{2.165}$$

or

$$a^* a = 1, \tag{2.166}$$

or

$$|a|^2 = 1. \tag{2.167}$$

Clearly, $a = \pm 1$. We choose $a = 1$, and, thus, we have that

$$\begin{aligned} |\lambda_1\rangle &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= |0\rangle . \end{aligned} \tag{2.168}$$

In a similar way, we find out that

$$\begin{aligned} |\lambda_2\rangle &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= |1\rangle . \end{aligned} \tag{2.169}$$

We can now confirm, for Π_{λ_i} being the orthogonal projector to the subspace corresponding to the eigenvalue λ_i, that

$$\begin{aligned} \sigma_z &= \sum_{i=1}^{2} \lambda_i \Pi_{\lambda_i} \\ &= \sum_{i=1}^{2} \lambda_i \, |\lambda_i\rangle\langle\lambda_i| \\ &= |0\rangle\langle 0| - |1\rangle\langle 1| \end{aligned}$$

$$
\begin{aligned}
&= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \qquad (2.170)
\end{aligned}
$$

2.4 Computational Complexity Theory

Computational complexity theory is, perhaps, the most significant aspect of theoretical computer science. It incorporates a large number of the basic paradigms on which much of the other subdisciplines of theoretical computer science rely on, and, also, contains a whealth of fundamental results and open problems.

A *decision problem* is a problem whose answer is either a `Yes`, or a `No`. We start by giving some definitions about some types of decision problems, and functions, that we will encounter. We fix the set $\Sigma = \{0, 1\}$ to be our alphabet. Note that the set $\Sigma^* = \bigcup_i^\infty A_i$, is the set that contains all of the finite strings that can be created by combining the elements of the set Σ.

Definition 2.54 (Promise Problems [103])**.** A *promise problem* is a pair $A = (A_{\text{yes}}, A_{\text{no}})$, where $A_{\text{yes}}, A_{\text{no}} \subseteq \Sigma^*$, such that $A_{\text{yes}} \cap A_{\text{no}} = \emptyset$. The set A_{yes} contains the *yes-instances* of the problem, that is, the instances that have answer `Yes`, whereas the set A_{no} contains the *no-instances* of the problem, that is, the instances that have answer `No`. Here, $A_{\text{yes}} \cup A_{\text{no}} \subseteq \Sigma^*$.

Definition 2.55 (Language Problems [103])**.** A *language problem* is a special case of a promise problem $A = (A_{\text{yes}}, A_{\text{no}})$, in which $A_{\text{yes}} \cup A_{\text{no}} = \Sigma^*$. That is, in a language problem the union of the yes-instances and the no-instances of the problem, forms a partition of Σ^*.

The following type of functions, will be useful, later, when we will be defining some complexity classes.

Definition 2.56 (Polynomial-Bounded Functions [103])**.** A function of the form $p : \mathbb{N} \to \mathbb{N}$, is said to be *polynomial-bounded function* if, and only if, there exists a polynomial-time deterministic Turing machine, that outputs

$1^{f(n)}$, on input 1^n, for every $n \in \mathbb{N}$. Such functions are upper-bounded by some polynomial, and are efficiently computable.

2.4.1 Classical Computational Complexity Theory

Complexity theory, in the classical setting, spans nearly fifty years of interesting results, regarding the exploration of the nature, and the paradigms, of computation.

Computational Models

In the classical setting, problems are solved by variations of the Turing machine, whose basic structure was discovered by Alan M. Turing in the thirties.

Turing machines are abstract idealized machines that their basic functions are similar to those of an actual computer. These machines, have access to some one-dimensional tape, which is infinite at its one direction, and are able to read, and write, symbols from, and on, this tape. Thus, by reading what is written on the tape, and conditionally writing upon it, Turing machines are able to do all of the stuff a modern computer can do. The processes of reading and writing are carried out by a "head," which is able to move right and left on the tape. Finally, Turing machines are somehow able to alter their internal state. This helps them distinguish among different phases of the computation, at hand, that is being carried out.

Definition 2.57 (Deterministic Turing Machines [80])**.** Formally, a *deterministic Turing machine* is a quadruple

$$M = (K, \Sigma, \delta, s)\,.$$

Here, K is a finite set of states, $s \in K$ is the initial state, and Σ is a finite set of symbols. We have that $K \cap \Sigma = \emptyset$, and that $\sqcup, \triangleright \in \Sigma$. We call $\sqcup$ the blank symbol, and $\triangleright$ the first symbol, respectively. Finally, δ is a transition function, or, in more formal terms,

$$\delta : K \times \Sigma \to (K \cup \{q_h, q_{\texttt{Yes}}, q_{\texttt{No}}\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\}\,. \tag{2.171}$$

Note that q_h is a halting state, $q_{\texttt{Yes}}$ is an accepting state, and $q_{\texttt{No}}$ is a rejecting state. The symbols of the set $\{\leftarrow, \rightarrow, -\}$ refer to the movement of the read-write head of the machine. Also,

$$(K \cup \Sigma) \cap \{\leftarrow, \rightarrow, -\} = \emptyset.$$

Definition 2.58 (Non-Deterministic Turing Machines [80]). Formally, a *non-deterministic Turing machine* is a quadruple

$$M = (K, \Sigma, \Delta, s)\,.$$

Here, K is a finite set of states, $s \in K$ is the initial state, and Σ is a finite set of symbols. We have that $K \cap \Sigma = \emptyset$, and that $\sqcup, \triangleright \in \Sigma$. We call $\sqcup$ the blank symbol, and $\triangleright$ the first symbol, respectively. Finally, Δ is a transition relation, or, in more formal terms,

$$\Delta \subseteq (K \times \Sigma) \times ((K \cup \{q_h, q_{\texttt{Yes}}, q_{\texttt{No}}\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\})\,. \tag{2.172}$$

Note that q_h is a halting state, $q_{\texttt{Yes}}$ is an accepting state, and $q_{\texttt{No}}$ is a rejecting state. The symbols of the set $\{\leftarrow, \rightarrow, -\}$ refer to the movement of the read-write head of the machine. Also,

$$(K \cup \Sigma) \cap \{\leftarrow, \rightarrow, -\} = \emptyset.$$

Definition 2.59 (Probabilistic Turing Machines [80, 87]). Formally, a *probabilistic Turing machine* is a quintuple

$$M = (K, \Sigma, \delta_0, \delta_1, s)\,.$$

Here, K is a finite set of states, $s \in K$ is the initial state, and Σ is a finite set of symbols. We have that $K \cap \Sigma = \emptyset$, and that $\sqcup, \triangleright \in \Sigma$. We call $\sqcup$ the blank symbol, and $\triangleright$ the first symbol, respectively. Finally, δ_i, for every i in $\{0, 1\}$, is a transition function, or, in more formal terms,

$$\forall i \in \{0,1\} : \delta_i : K \times \Sigma \to (K \cup \{q_h, q_{\texttt{Yes}}, q_{\texttt{No}}\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\}\,. \tag{2.173}$$

Note that q_h is a halting state, $q_{\texttt{Yes}}$ is an accepting state, and $q_{\texttt{No}}$ is a rejecting state. The symbols of the set $\{\leftarrow, \rightarrow, -\}$ refer to the movement of the read-write head of the machine. Also,

$$(K \cup \Sigma) \cap \{\leftarrow, \rightarrow, -\} = \emptyset.$$

In each step, the machine flips a binary coin c. If $c = i \in \{0, 1\}$, then M uses δ_i, as a transition function, for the current step of the computation, being carried out.

Note 2.60. The deterministic Turing machines are a special case of the probabilistic Turing machines, since one can set

$$\delta_0 = \delta_1 = \delta.$$

Also, probabilistic Turing machines are a special case of non-deterministic Turing machines, since one can set

$$\Delta = \delta_0 \cup \delta_1,$$

if we perceive the functions $\delta_i : A \to B$ as binary, input-output in nature, relations $\delta_i \subseteq A \times B$, for every i in the set $\{0, 1\}$.

Fundamental Classical Complexity Classes

We present the classical computational complexity classes P, BPP, NP, MA, MA$_1$, BPP$_{\mathsf{PATH}}$, PP, PSPACE, NPSPACE, and EXP.

Definition 2.61 (The Complexity Class P [103]). A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in P if, and only if, there exists a polynomial-time deterministic machine M, such that the machine M accepts every string $x \in A_{\text{yes}}$, and rejects every string $x \in A_{\text{no}}$.

Definition 2.62 (The Complexity Class BPP [103]). A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in BPP if, and only if, there exists a polynomial-time probabilistic Turing machine M, such that M accepts every string $x \in A_{\text{yes}}$, with probability at least $\frac{2}{3}$, and accepts every string $x \in A_{\text{no}}$, with probability at most $\frac{1}{3}$.

Definition 2.63 (The Complexity Class NP [103]). A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in NP if, and only if, there exists a polynomial-bounded function p, and a polynomial-time deterministic Turing machine M, with the following properties. For every string $x \in A_{\text{yes}}$, it holds that M accepts (x, y), for some string $y \in \Sigma^{p(|x|)}$, and for every string $x \in A_{\text{no}}$, it holds that M rejects (x, y), for all strings $y \in \Sigma^{p(|x|)}$.

Definition 2.64 (The Complexity Class MA [103]). A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in MA if, and only if, there exists a polynomial-bounded function p, and a polynomial-time Turing machine M, with the following properties. For every string $x \in A_{\text{yes}}$, it holds that

$$\Pr\left[\text{The machine } M \text{ accepts } (x, y).\right] \geq \frac{2}{3},$$

for some string $y \in \Sigma^{p(|x|)}$, and, for every string $x \in A_{\text{no}}$, it holds that

$$\Pr\left[\text{The machine } M \text{ accepts } (x, y).\right] \leq \frac{1}{3},$$

for all strings $y \in \Sigma^{p(|x|)}$.

Definition 2.65 (The Complexity Class MA_1)**.** A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in MA_1 if, and only if, there exists a polynomial-bounded function p, and a polynomial-time Turing machine M, with the following properties. For every string $x \in A_{\text{yes}}$, it holds that

$$\Pr\left[\text{The machine } M \text{ accepts } (x, y).\right] = 1,$$

for some string $y \in \Sigma^{p(|x|)}$, and, for every string $x \in A_{\text{no}}$, it holds that

$$\Pr\left[\text{The machine } M \text{ accepts } (x, y).\right] \leq \frac{1}{3},$$

for all strings $y \in \Sigma^{p(|x|)}$.

Definition 2.66 (The Complexity Class $\mathsf{BPP}_{\mathsf{PATH}}$ [57])**.** The class $\mathsf{BPP}_{\mathsf{PATH}}$ is the class of language problems $L \subseteq \Sigma^*$, for which there exists a BPP-machine M, such that M can either "succeed" or "fail," and, conditioned on succeeding, can either accept or reject. We have that, for all inputs x,

- $\Pr\left[\text{The computation } M(x) \text{ succeeds.}\right] > 0$,
- if $x \in L$, then

$$\Pr\left[\text{The computation } M(x) \text{ accepts.} \mid \text{The computation } M(x) \text{ succeeds.}\right] \geq \frac{2}{3}.$$

- and, if $x \notin L$, then

$$\Pr\left[\text{The computation } M(x) \text{ accepts.} \mid \text{The computation } M(x) \text{ succeeds.}\right] \leq \frac{1}{3}.$$

Note 2.67. In the class $\mathsf{BPP}_{\mathsf{PATH}}$, we have the ability to "post-select." That is, we can define our own notion of "success," in order for it to be a desired, for us, property to have, regarding our current problem at hand. Also, note that the symbol $\Pr[B \mid A]$ denotes "the probability that the event B occurs, given that the event A occurs." Note that

$$\Pr[B \mid A] = \frac{\Pr[B \cap A]}{\Pr[A]}. \tag{2.174}$$

Definition 2.68 (The Complexity Class PP [103])**.** A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in PP if, and only if, there exists a polynomial-time probabilistic Turing machine M, such that M accepts every string $x \in A_{\text{yes}}$, with probability strictly greater than $\frac{1}{2}$, and accepts every string $x \in A_{\text{no}}$, with probability at most $\frac{1}{2}$.

Definition 2.69 (The Complexity Class PSPACE [103])**.** A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in PSPACE if, and only if, there exists a deterministic Turing machine M, running in polynomial space, such that M accepts every string $x \in A_{\text{yes}}$, and rejects every string $x \in A_{\text{no}}$.

Definition 2.70 (The Complexity Class NPSPACE [80])**.** A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in NPSPACE if, and only if, there exists a non-deterministic Turing machine M, running in polynomial space, such that M accepts every string $x \in A_{\text{yes}}$, and rejects every string $x \in A_{\text{no}}$.

Definition 2.71 (The Complexity Class EXP [103])**.** A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in EXP if, and only if, there exists a deterministic Turing machine M, running in exponential time, that is, in time bounded by 2^p, where p is some polynomial-bounded function, such that M accepts every string $x \in A_{\text{yes}}$, and rejects every string $x \in A_{\text{no}}$.

2.4.2 Quantum Computational Complexity Theory

The projection of complexity theory in quantum mechanics yields quantum complexity theory.

Computational Models

In the quantum setting, problems are usually solved by quantum circuits. Quantum circuits are a generalization of classical, Boolean, circuits. A Boolean circuit can be inspected in Figure 2.2. Note that every Boolean circuit, can be composed by NOT and AND gates only.

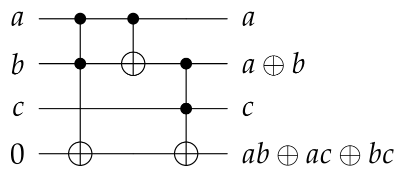

Figure 2.2: A circuit implementation, of some algorithm that solves the problem `MAJORITY`, which asks whether a given binary string, of length three, here, has more ones than zeros. In this implementation, only classical CNOT, and CCNOT, gates are used. That is, controlled-NOT and controlled-controlled-NOT gates.

Definition 2.72 (Quantum Circuits)**.** *Quantum circuits* are like ordinary Boolean circuits, but, in contrast, they operate on qubits instead of bits. Now, every quantum circuit can be composed by combining gates from the set

$$\{\text{CNOT}, H, T\},$$

where

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \tag{2.175}$$

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{2.176}$$

and

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{j\frac{\pi}{4}} \end{pmatrix}, \tag{2.177}$$

where $j^2 = -1$. Note that CNOT acts on two qubits, while H and T act on one qubit. Also, not that all the members of the set $\{\text{CNOT}, H, T\}$ are unitary transformations, that act on complex vector spaces, which contain the vectors that hold the one-qubit, and two-qubit, quantum states.

An example of a quantum circuit, can be inspected in Figure 2.3.

Note 2.73 (Quantum Turing Machines)**.** As it turns out, there is a quantum version of the Turing machine computational model, as well. However, we

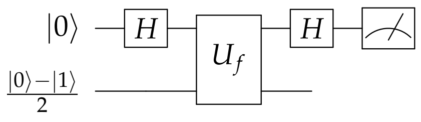

Figure 2.3: A quantum circuit implementation, of some algorithm that solves the `DEUTSCH PROBLEM`. In the `DEUTSCH PROBLEM`, we are given a Boolean function $f : \{0,1\} \to \{0,1\}$, and we are asked to compute $f(0) \oplus f(1)$. Note that U_f denotes the, unitary, in nature, oracle for computing the values of f. The unitary U_f acts like this: $U_f |x\rangle |b\rangle = |x\rangle |y \oplus f(x)\rangle$.

are not going to elaborate further on this model, here, as the quantum circuit is adequate enough to define all of the quantum complexity classes that we need in this work. For a reference, one can see the work by Bernstein and Vazirani [43], or the work by Deutsch [36].

Fundamental Quantum Complexity Classes

We present the quantum computational complexity classes BQP, QMA, QMA_1, SQMA, SQMA_1, QCMA, QCMA_1, and $\mathsf{QCMA}_{\mathsf{EXP}}$.

Definition 2.74 (The Complexity Class BQP [103])**.** Let $A = (A_{\text{yes}}, A_{\text{no}})$ be a promise problem, and let $a, b : \mathbb{N} \to [0,1]$ be functions. Then, $A \in \mathsf{BQP}(a,b)$ if, and only if, there exists a polynomial-time generated family of quantum circuits $\{Q_n\}_n$, where each circuit Q_n takes n input qubits, and produces one output qubit, that satisfies the following proprerties:

- if $x \in A_{\text{yes}}$, then $\Pr[\text{The circuit } Q_{|x|} \text{ accepts.}] \geq a(|x|)$, and
- if $x \in A_{\text{no}}$, then $\Pr[\text{The circuit } Q_{|x|} \text{ accepts.}] \leq b(|x|)$.

The class BQP is defined as $\mathsf{BQP} = \mathsf{BQP}\left(\frac{2}{3}, \frac{1}{3}\right)$.

Definition 2.75 (The Complexity Class QMA [103])**.** Let $A = (A_{\text{yes}}, A_{\text{no}})$ be a promise problem, let $p(n)$ be a polynomial-bounded function, and let $a, b : \mathbb{N} \to [0,1]$ be functions. Then, $A \in \mathsf{QMA}_p(a,b)$ if, and only if, there exists a polynomial-time generated family of quantum circuits $\{Q_n\}_n$, where each circuit Q_n takes $n + p(n)$ input qubits, and produces one output qubit, with the following properties.

- *Completeness.* For all $x \in A_{\text{yes}}$, there exists a $p(|x|)$-qubit quantum state $|\psi\rangle$, such that

$$\Pr\left[\text{The circuit } Q_{|x|} \text{ accepts } (x, |\psi\rangle).\right] \geq a(|x|).$$

- *Soundness.* For all $x \in A_{\text{no}}$, and all $p(|x|)$-qubit quantum states $|\psi\rangle$, it is the case that

$$\Pr\left[\text{The circuit } Q_{|x|} \text{ accepts } (x, |\psi\rangle).\right] \leq b(|x|).$$

Also, define $\mathsf{QMA} = \bigcup_p \mathsf{QMA}_p\left(\frac{2}{3}, \frac{1}{3}\right)$, where the union is over all polynomial-bounded functions p.

Definition 2.76 (The Complexity Class QMA_1)**.** Let $A = \left(A_{\text{yes}}, A_{\text{no}}\right)$ be a promise problem, let $p(n)$ be a polynomial-bounded function, and let $a, b : \mathbb{N} \to [0,1]$ be functions. Then, $A \in \mathsf{QMA}_p(a,b)$ if, and only if, there exists a polynomial-time generated family of quantum circuits $\{Q_n\}_n$, where each circuit Q_n takes $n + p(n)$ input qubits, and produces one output qubit, with the following properties.

- *Completeness.* For all $x \in A_{\text{yes}}$, there exists a $p(|x|)$-qubit quantum state $|\psi\rangle$, such that

$$\Pr\left[\text{The circuit } Q_{|x|} \text{ accepts } (x, |\psi\rangle).\right] \geq a(|x|).$$

- *Soundness.* For all $x \in A_{\text{no}}$, and all $p(|x|)$-qubit quantum states $|\psi\rangle$, it is the case that

$$\Pr\left[\text{The circuit } Q_{|x|} \text{ accepts } (x, |\psi\rangle).\right] \leq b(|x|).$$

Also, define $\mathsf{QMA}_1 = \bigcup_p \mathsf{QMA}_p\left(1, \frac{1}{3}\right)$, where the union is over all polynomial-bounded functions p.

Definition 2.77 (Subset-States [55])**.** Let $\mathcal{Q}$ be a quantum system on n qubits. Then, the dimension of the vector space $\mathcal{H}$, which holds the vectors $|\psi\rangle$ that describe the state of $\mathcal{Q}$, is 2^n. For any subset $S \subseteq [2^n]$, we define the quantum state

$$|S\rangle = \sum_{i \in S} \frac{1}{\sqrt{|S|}} |i\rangle \tag{2.178}$$

to be a *subset-state*. Note that $|S\rangle$ is a uniform superposition over the elements of the computational basis that their binary labels, after converted to integers, belong to S. Note, also, that every subset-state is a just a special case of a pure state, and that

$$\begin{aligned} \| |S\rangle \|_2^2 &= \left(\sqrt{\sum_{i=1}^{|S|} \left| \frac{1}{\sqrt{|S|}} \right|^2} \right)^2 \\ &= \frac{|S|}{|S|} \\ &= 1. \end{aligned} \tag{2.179}$$

Definition 2.78 (The Complexity Class SQMA)**.** Let $A = (A_{\text{yes}}, A_{\text{no}})$ be a promise problem, let $p(n)$ be a polynomial-bounded function, and let $a, b : \mathbb{N} \to [0,1]$ be functions. Then, $A \in \mathsf{SQMA}_p(a,b)$ if, and only if, there exists a polynomial-time generated family of quantum circuits $\{Q_n\}_n$, where each circuit Q_n takes $n + p(n)$ input qubits, and produces one output qubit, with the following properties.

- *Completeness.* For all $x \in A_{\text{yes}}$, there exists a $p(|x|)$-qubit quantum subset-state $|S\rangle$, with $S \subseteq \left[2^{p(|x|)}\right]$, such that

$$\Pr\left[\text{The circuit } Q_{|x|} \text{ accepts } (x, |S\rangle).\right] \geq a(|x|).$$

- *Soundness.* For all $x \in A_{\text{no}}$, and all $p(|x|)$-qubit quantum subset-states $|S\rangle$, with $S \subseteq \left[2^{p(|x|)}\right]$, it is the case that

$$\Pr\left[\text{The circuit } Q_{|x|} \text{ accepts } (x, |S\rangle).\right] \leq b(|x|).$$

Also, define $\mathsf{SQMA} = \bigcup_p \mathsf{SQMA}_p\left(\frac{2}{3}, \frac{1}{3}\right)$, where the union is over all polynomial-bounded functions p.

Definition 2.79 (The Complexity Class SQMA_1)**.** Let $A = (A_{\text{yes}}, A_{\text{no}})$ be a promise problem, let $p(n)$ be a polynomial-bounded function, and let $a, b : \mathbb{N} \to [0,1]$ be functions. Then, $A \in \mathsf{SQMA}_p(a,b)$ if, and only if, there exists a polynomial-time generated family of quantum circuits $\{Q_n\}_n$, where each circuit Q_n takes $n + p(n)$ input qubits, and produces one output qubit, with the following properties.

- *Completeness.* For all $x \in A_{\text{yes}}$, there exists a $p(|x|)$-qubit quantum subset-state $|S\rangle$, with $S \subseteq \left[2^{p(|x|)}\right]$, such that

$$\Pr\left[\text{The circuit } Q_{|x|} \text{ accepts } (x, |S\rangle).\right] \geq a(|x|).$$

- *Soundness*. For all $x \in A_{\text{no}}$, and all $p(|x|)$-qubit quantum subset-states $|S\rangle$, with $S \subseteq \left[2^{p(|x|)}\right]$, it is the case that

$$\Pr\left[\text{The circuit } Q_{|x|} \text{ accepts } (x, |S\rangle).\right] \leq b(|x|).$$

Also, define $\mathsf{SQMA}_1 = \bigcup_p \mathsf{SQMA}_p\left(1, \frac{1}{3}\right)$, where the union is over all polynomial-bounded functions p.

Definition 2.80 (The Complexity Class QCMA [103]). Let $A = (A_{\text{yes}}, A_{\text{no}})$ be a promise problem. Then, $A \in \mathsf{QCMA}$ if, and only if, there exists a polynomial-bounded function $p(n)$, and a polynomial-time generated family of quantum circuits $\{Q_n\}_n$, where each circuit Q_n takes $n + p(n)$ input qubits, and produces one output qubit, with the following properties. For all of the inputs $x \in A_{\text{yes}}$, there is a string $y \in \Sigma^{p(n)}$, such that

$$\Pr\left[\text{The circuit } Q_{|x|+p(|x|)} \text{ accepts } (x, y).\right] \geq \frac{2}{3},$$

and, for all of the inputs $x \in A_{\text{no}}$, and all of the strings $y \in \Sigma^{p(n)}$,

$$\Pr\left[\text{The circuit } Q_{|x|+p(|x|)} \text{ accepts } (x, y).\right] \leq \frac{1}{3}.$$

Note 2.81 (Names for QCMA). The class QCMA, defined by Aharonov and Naveh in 2002 [17], can be also encountered as CMQA, or MQA [103].

Definition 2.82 (The Complexity Class QCMA_1). Let $A = (A_{\text{yes}}, A_{\text{no}})$ be a promise problem. Then, $A \in \mathsf{QCMA}_1$ if, and only if, there exists a polynomial-bounded function $p(n)$, and a polynomial-time generated family of quantum circuits $\{Q_n\}_n$, where each circuit Q_n takes $n + p(n)$ input qubits, and produces one output qubit, with the following properties. For all of the inputs $x \in A_{\text{yes}}$, there is a string $y \in \Sigma^{p(n)}$, such that

$$\Pr\left[\text{The circuit } Q_{|x|+p(|x|)} \text{ accepts } (x, y).\right] = 1,$$

and, for all of the inputs $x \in A_{\text{no}}$, and all of the strings $y \in \Sigma^{p(n)}$,

$$\Pr\left[\text{The circuit } Q_{|x|+p(|x|)} \text{ accepts } (x, y).\right] \leq \frac{1}{3}.$$

Definition 2.83 (The Complexity Class $\mathsf{QCMA}_{\mathsf{EXP}}$ [103])**.** Let $A = \left(A_{\text{yes}}, A_{\text{no}}\right)$ be a promise problem. Then, $A \in \mathsf{QCMA}_{\mathsf{EXP}}$ if, and only if, there exists a polynomial-bounded function $p\left(n\right)$, and a exponential-time generated family of quantum circuits $\left\{Q_n\right\}_n$, where each circuit Q_n takes $n + p\left(n\right)$ input qubits, and produces one output qubit, with the following properties. For all of the inputs $x \in A_{\text{yes}}$, there is a string $y \in \Sigma^{p(n)}$, such that

$$\Pr\left[\text{The circuit } Q_{|x|+p(|x|)} \text{ accepts } (x, y).\right] \geq \frac{2}{3},$$

and, for all of the inputs $x \in A_{\text{no}}$, and all of the strings $y \in \Sigma^{p(n)}$,

$$\Pr\left[\text{The circuit } Q_{|x|+p(|x|)} \text{ accepts } (x, y).\right] \leq \frac{1}{3}.$$

2.4.3 Relativized Worlds

What is a relativized world? Well, it is a world induced by calls to some *oracle*, be it a *classical* or a *quantum* one. In these worlds, we work with oracle machines. Oracle machines are ordinary computational machines, like Turing machines, say, which are equipped with a subroutine that, in unit time-cost, solves a highly non-trivial, and "difficult," problem. This oracle is appropriately selected, in order for the desired separation to be carried out effectively.

Definition 2.84 (Turing Machines with Oracles [80, 87])**.** Formally, a *Turing machine with an oracle* $\mathcal{A}$ is a quadruple

$$M^{\mathcal{A}} = \left(K, \Sigma, \delta, s\right).$$

Here, K is a finite set of states, $s \in K$ is the initial state, and Σ is a finite set of symbols. We have that $K \cap \Sigma = \emptyset$, and that $\sqcup, \rhd \in \Sigma$. We call $\sqcup$ the blank symbol, and $\rhd$ the first symbol, respectively. Finally, δ is a transition function, or, in more formal terms,

$$\delta : K \times \Sigma \to \left(K \cup \left\{q_h, q_{\texttt{Yes}}, q_{\texttt{No}}\right\}\right) \times \Sigma \times \left\{\leftarrow, \rightarrow, -\right\}. \tag{2.180}$$

Note that q_h is a halting state, $q_{\texttt{Yes}}$ is an accepting state, and $q_{\texttt{No}}$ is a rejecting state. The symbols of the set $\left\{\leftarrow, \rightarrow, -\right\}$ refer to the movement of the read-write head of the machine. Also,

$$\left(K \cup \Sigma\right) \cap \left\{\leftarrow, \rightarrow, -\right\} = \emptyset.$$

Finally, there are some special-purpose states $q_{\texttt{query}}, q_{\texttt{query-Yes}}, q_{\texttt{query-No}} \in K$. There is an extra query tape, in which the machine can write a string x, of which the membership in the set $\mathcal{A}$ wants to decide, and gets into the query state $q_{\texttt{query}}$. Then, in one time-step, the machine enters the state $q_{\texttt{query-Yes}}$, or $q_{\texttt{query-No}}$, regarding whether $x \in \mathcal{A}$, or $x \notin \mathcal{A}$, respectively. The fact that the machine $M^{\mathcal{A}}$ decides the membership in the set $\mathcal{A}$, in only one step, is a magical imaginary, non-trivial, property of $M^{\mathcal{A}}$.

Note 2.85. Turing machines with oracles are defined in a similar manner, when it comes to non-determinism, or probabilism.

Classical Oracles

Let $n \in \mathbb{N}$. Classical oracles implement some Boolean function f which is defined on strings drawn from the set $\{0,1\}^n$. That is, they implement Boolean functions of the form

$$f : \{0,1\}^n \to \{0,1\} . \tag{2.181}$$

Classical Oracles In a Classical Setting. Classical oracles, in a classical setting, are functions U_f such that

$$U_f : x \mapsto f(x) . \tag{2.182}$$

Classical Oracles In a Quantum Setting. Classical oracles, in a quantum setting, are unitary transformations U_f such that, for $b \in \{0,1\}$,

$$U_f : |x\rangle |b\rangle \mapsto |x\rangle |b \oplus f(x)\rangle , \tag{2.183}$$

or

$$U_f |x\rangle |b\rangle = |x\rangle |b \oplus f(x)\rangle . \tag{2.184}$$

Equivalently, a classical oracle, in a quantum setting, can be sought as a quantum transformation U_f, such that

$$U_f : |x\rangle \mapsto (-1)^{f(x)} |x\rangle , \tag{2.185}$$

or

$$U_f |x\rangle = (-1)^{f(x)} |x\rangle . \tag{2.186}$$

Another equivalent form, of the Equations (2.185) and (2.186), is to implement some Boolean function

$$f : \{0,1\}^n \to \{-1,1\}\,, \tag{2.187}$$

as

$$U_f : |x\rangle \mapsto f(x)\,|x\rangle\,, \tag{2.188}$$

or

$$U_f\,|x\rangle\,|b\rangle = f(x)\,|x\rangle\,. \tag{2.189}$$

The use of classical oracles. We can create families of classical oracles, like

$$\begin{aligned}\mathcal{F} &= \{f_1, f_2, \ldots, f_n, \ldots\} \\ &= \left\{f_n \mid f_n : \{0,1\}^n \to \{0,1\}\right\}_{n\in\mathbb{N}} \\ &= \mathcal{F}_{\text{good}} \cup \mathcal{F}_{\text{bad}} \\ &= \mathcal{F}_{\text{good}} \uplus \mathcal{F}_{\text{bad}} \\ &= \left\{f_n \mid \exists x \in \{0,1\}^n : f_n(x) = 1\right\}_{n\in\mathbb{N}} \\ &\qquad \uplus \left\{f_n \mid \forall x \in \{0,1\}^n : f_n(x) = 0\right\}_{n\in\mathbb{N}}\,. \end{aligned} \tag{2.190}$$

The problem put forth, here, is to decide whether a given function f_k, for some natural $k \in \mathbb{N}$, is either good or bad—as defined above.

Quantum Oracles

These oracles can be viewed either as unitaries, or CPTP maps.

Oracles as Unitaries. Quantum oracles, are unitary transformations that implement an unknown unitary transformation, namely U. One example, put forth [13], is the case of quantum oracles that are capable of identifying an unknown state $|\psi\rangle$, that is,

$$U_{|\psi\rangle} : |\psi\rangle \mapsto -\,|\psi\rangle\,, \tag{2.191}$$

or

$$U_f\,|\psi\rangle = -\,|\psi\rangle\,, \tag{2.192}$$

but

$$U_{|\psi\rangle} : |\phi\rangle \mapsto |\phi\rangle\,, \tag{2.193}$$

or

$$U_f\,|\psi\rangle = |\psi\rangle\,, \tag{2.194}$$

for every $|\phi\rangle$, such that $\langle\phi|\psi\rangle = 0$. In this case, the application of the oracle to some other state $|\varphi\rangle$, that is neither equal to $|\psi\rangle$ nor is it orthogonal to the ket $|\psi\rangle$, is arbitrary.

Using these oracles, which conceal quantum states, we can create generic oracle-sets of the form

$$\begin{aligned}\mathcal{U} &= \{U_i\}_{i\in\mathbb{N}}\\ &= \left\{U_i \mid U_i = U_{|\psi\rangle}\right\}_{i\in\mathbb{N}} \cup \{U_i \mid U_i = I\}_{i\in\mathbb{N}}\\ &= \left\{U_i \mid U_i = U_{|\psi\rangle}\right\}_{i\in\mathbb{N}} \uplus \{U_i \mid U_i = I\}_{i\in\mathbb{N}}\\ &= \mathcal{U}_{\text{good}} \uplus \mathcal{U}_{\text{bad}}.\end{aligned} \tag{2.195}$$

That is, for every i, either U_i is good, or

$$U_i = U_{|\psi\rangle}, \tag{2.196}$$

for some state $|\psi\rangle$, or U_i is bad, or

$$U_i = I. \tag{2.197}$$

In the case of unitary quantum oracles, the language problem associated with the separation is this: given an index $n \in \mathbb{N}$, as the unary 1^n, and some purported proof $|\psi\rangle$, decide whether $U_n \in \mathcal{U}_{\text{good}}$, or $U_n \in \mathcal{U}_{\text{bad}}$. The purported proof is about the claim that $U_n \in \mathcal{U}_{\text{good}}$, put forth by the prover.

Oracles as CPTP Maps. Another case of quantum oracles, is when we employ CPTP maps, rather than unitaries. In these cases, we have the family of CPTP maps $\mathscr{U}_i$

$$\begin{aligned}\mathcal{U} &= \{\mathscr{U}_i\}_{i\in\mathbb{N}}\\ &= \{\mathscr{U}_i \mid \mathscr{U}_i \text{ is good}\}_{i\in\mathbb{N}} \cup \{\mathscr{U}_i \mid \mathscr{U}_i \text{ is bad}\}_{i\in\mathbb{N}}\\ &= \{\mathscr{U}_i \mid \mathscr{U}_i \text{ is good}\}_{i\in\mathbb{N}} \uplus \{\mathscr{U}_i \mid \mathscr{U}_i \text{ is bad}\}_{i\in\mathbb{N}}\\ &= \mathcal{U}_{\text{good}} \uplus \mathcal{U}_{\text{bad}},\end{aligned} \tag{2.198}$$

where $\mathcal{U}_{\text{good}}$ denotes the set of good CPTP maps, in an appropriate sense, defined to be different every time, and $\mathcal{U}_{\text{bad}}$ denotes the set of the bad ones, also defined differently each time.

In the cases of CPTP quantum oracles, the language problem associated with the separation is this: given an index $n \in \mathbb{N}$, as the unary 1^n, and some purported proof $|\psi\rangle$, decide whether $\mathscr{U}_n \in \mathcal{U}_{\text{good}}$, or $\mathscr{U}_n \in \mathcal{U}_{\text{bad}}$. The purported proof is about the claim that $\mathscr{U}_n \in \mathcal{U}_{\text{good}}$, put forth by the prover.

Note 2.86. Note that quantum oracles are meaningful only in a quantum setting. The reason is that the invocation of some quantum oracle requires quantum entities like quantum states to be applied on.

2.4.4 Fundamental Results

In this subsection we present some known inclusions about some complexity classes of interest, be them classical or quantum.

Theorem 2.87. We have that

$$\begin{aligned}
\mathsf{P} &\subseteq_{(1)} \mathsf{BPP} \\
&\subseteq_{(2)} \mathsf{BQP} \\
&\subseteq_{(3a)} \mathsf{QCMA} =_{(3b)} \mathsf{QCMA}_1 \\
&\subseteq_{(4)} \mathsf{SQMA}_1 \\
&\subseteq_{(5)} \mathsf{QMA}_1 \\
&\subseteq_{(6a)} \mathsf{QMA} =_{(6b)} \mathsf{SQMA} \\
&\subseteq_{(7)} \mathsf{PP} \\
&\subseteq_{(8a)} \mathsf{PSPACE} =_{(8b)} \mathsf{NPSPACE} \\
&\subseteq_{(9)} \mathsf{EXP}, \\
\mathsf{P} &\subseteq_{(10)} \mathsf{NP} \\
&\subseteq_{(11a)} \mathsf{MA} =_{(11b)} \mathsf{MA}_1 \\
&\subseteq_{(12)} \mathsf{QCMA}, \\
\mathsf{BPP} &\subseteq_{(13)} \mathsf{MA}, \\
\mathsf{MA} &\subseteq_{(14)} \mathsf{BPP}_{\mathsf{PATH}}, \\
\mathsf{BPP}_{\mathsf{PATH}} &\subseteq_{(15)} \mathsf{PP},
\end{aligned}$$

and

$$\mathsf{EXP} \subseteq_{(16)} \mathsf{QCMA}_{\mathsf{EXP}}.$$

Proof. Below, we discuss each inclusion separately.

$\subseteq_{(1)}$: The class P is a subset of the class BPP, since every deterministic algorithm can be sought as a special case of a probabilistic algorithm that does not make use of its access to randomness.

$\subseteq_{(2)}$: The class BPP is a subset of the class BQP, since every probabilistic algorithm can be sought as a special case of a quantum algorithm. The reason is that quantum computers have access to randomness, so, they can efficiently simulate any BPP-computer.

$\subseteq_{(3a)}$: The class BQP is a subset of the class QCMA, since one can give the polynomial-time algorithm, that generates the members of the uniform family of quantum circuits, for some BQP language, as a witness, or proof, and, then, a quantum computer can run the appropriate quantum circuit, after it generates it in polynomial time, and reply according to the result of this circuit. Note that the length of the classical witness, here, is polynomially big: every polynomial-time algorithm has polynomially-big classical description.

$=_{(3b)}$: This result comes from the relevant work by Jordan, Kobayashi, Nagaj, and Nishimura [63].

$\subseteq_{(4)}$: The class QCMA_1 is a subset of the class SQMA_1, since every classical witness can be encoded as a subset state, which consists of only one ket: the ket that corresponds to the binary representation of the classical witness. This is always possible since every classical witness, that is, every bit-string, can be matched to some element of the computational basis.

$\subseteq_{(5)}$: The class SQMA_1 is a subset of the class QMA_1, since every subset-state can be sought as a pure state. That is, every subset-state is a special case of a pure state.

$\subseteq_{(6a)}$: The class QMA_1 is a subset of the class QMA, since the property of perfect completeness is an extra, special, feature of the QMA verification protocols.

$=_{(6b)}$: The class SQMA is equal to the class QMA, by the work of Grilo, Kerenidis, and Sikora [55].

$\subseteq_{(7)}$: The class QMA is a subset of the class PP, by the work of Marriott, and Watrous [74].

$\subseteq_{(8a)}$: The class PP is a subset of the class PSPACE, since, in polynomial space, one can count all of the `Yes`-paths of the Turing machine, at hand, and, then, decide whether these paths are more than half of the entire number of computational paths.

$=_{(8b)}$: We know that PSPACE is a subset of NPSPACE, by the definitions of these classes. The fact that NPSPACE is a subset of PSPACE, comes from Savitch's theorem [80].

$\subseteq_{(9)}$: The class PSPACE is a subset of the class EXP, since, in exponential time, one can fully simulate any PSPACE-machine. The reason is that a PSPACE-machine can only have exponentially-many different configurations.

$\subseteq_{(10)}$: The class P is a subset of the class NP, since every deterministic Turing machine can be sought as a special case of a non-deterministic one.

$\subseteq_{(11a)}$: The class NP is a subset of the class MA, since the deterministic verification of the purported proof, that arises in the class NP, is a special case of a probabilistic verification. Probabilistic verifications, define the class MA.

$=_{(11b)}$: The class MA is equal to the class MA_1, from a result by Zachos and Fürer [106].

$\subseteq_{(12)}$: The class MA is a subset of the class QCMA, since any probabilistic verifier can be sought as a special case of some quantum verifier.

$\subseteq_{(13)}$: The class BPP is a subset of the class MA, since one can ignore the purported proof, of the MA setting, and just make use of the probabilistic polynomial time to solve the BPP problem.

$\subseteq_{(14)}$: The class MA is a subset of the class $\mathsf{BPP}_{\mathsf{PATH}}$, since one can post-select on getting a valid proof, and then make use of the probabilistic polynomial time to verify it.

$\subseteq_{(15)}$: The class $\mathsf{BPP}_{\mathsf{PATH}}$ is a subset of the class PP, from a result by Han, Hemaspaandra, and Thierauf [57].

$\subseteq_{(16)}$: The class EXP is a subset of the class $\mathsf{QCMA}_{\mathsf{EXP}}$, for the trivial reason that every $\mathsf{QCMA}_{\mathsf{EXP}}$-machine is allowed to run for exponentially-many time-steps.

$\square$

The inclusions of Theorem 2.87, can alternatively be inspected in the Hassel diagram, depicted in the Figure 2.4.

$$\begin{tikzcd}
 & \mathsf{QCMA}_{\mathsf{EXP}} & & \\
 & \mathsf{EXP} \arrow[u] & & \\
 & \mathsf{PSPACE} = \mathsf{NPSPACE} \arrow[u] & & \\
 & \mathsf{PP} \arrow[u] & & \\
 & \mathsf{QMA} = \mathsf{SQMA} \arrow[u] & & \cdot \arrow[ull] \\
 & \mathsf{QMA}_1 \arrow[u] & & \\
 & \mathsf{SQMA}_1 \arrow[u] & & \\
 & \mathsf{QCMA} = \mathsf{QCMA}_1 \arrow[u] & & \mathsf{BPP}_{\mathsf{PATH}} \arrow[uuu, dash] \\
\mathsf{BQP} \arrow[ur] & & \mathsf{MA} = \mathsf{MA}_1 \arrow[ul] \arrow[ur] & \\
\mathsf{BPP} \arrow[u] \arrow[urr] & & \mathsf{NP} \arrow[u] & \\
 & \mathsf{P} \arrow[ul] \arrow[ur] & &
\end{tikzcd}$$

Figure 2.4: A Hasse diagram depicting the known inclusions among some of the most frequently used complexity classes, in this work, be them classical or quantum. Each arrow of the form $\mathsf{A} \to \mathsf{B}$ illustrates the fact "$\mathsf{A} \subseteq \mathsf{B}$."

Chapter 3

Literature Review

That quantum mechanics should have implications to computational complexity theory, however, is much less clear. It is only through the remarkable discoveries and ideas of several researchers [...] that this potential has become evident.

—John Watrous,
Quantum Computational Complexity (2008)

We survey some of the most significant results about classical and quantum computational complexity classes, in various relativized worlds, induced by calls to classical or quantum oracles. We discriminate among three cases, regarding whether we examine a classical or a quantum oracle application, to the ground of classical or quantum complexity classes.

In the following sections we list, per section, a handful of somewhat influential, and important, results, according to the type of the oracle employed, and the complexity classes involved. Note that, in the following, ORACLES denotes the set of all oracles, that is, classical or quantum. Also, note that many of the following results might imply some of the other results. Nonetheless, we present them here as they were first published.

3.1 Classical Oracles in a Classical Setting

coNP is the class that contains the complements of the languages of NP. IP stands for *interactive proofs*.

1. $\exists\mathcal{A} \in \textsc{oracles} : \mathsf{NP}^{\mathcal{A}} \not\subseteq \mathsf{P}^{\mathcal{A}}$ [96].
2. $\exists\mathcal{A} : \mathsf{P}^{\mathcal{A}} \neq \mathsf{coNP}^{\mathcal{A}}$ [31].
3. $\exists\mathcal{A} : \mathsf{IP}^{\mathcal{A}} \neq \mathsf{PSPACE}^{\mathcal{A}}$ [30].

3.2 Classical Oracles in a Quantum Setting

In the following, EQP stands for *exact quantum polynomial-time*, and RP stands for *randomized polynomial-time*. coRP contains the complements of the languages of RP. ZPP is *zero-error probabilistic polynomial-time*. BQEXP is *bounded-error quantum exponential-time*. 2-EXP is *doubly-exponential-time*.

1. $\exists\mathcal{A} : \mathsf{NP}^{\mathcal{A}} \not\subseteq \mathsf{BQP}^{\mathcal{A}}$ [25].
2. $\exists\mathcal{A} : (\mathsf{NP} \cap \mathsf{coNP})^{\mathcal{A}} \not\subseteq \mathsf{BQP}^{\mathcal{A}}$ [25].
3. $\exists\mathcal{A} : \mathsf{BQP}^{\mathcal{A}} \not\subseteq \mathsf{BPP}^{\mathcal{A}}$ [93].
4. $\exists\mathcal{A} : \mathsf{BQP}^{\mathcal{A}} \not\subseteq \mathsf{BPP}^{\mathcal{A}}_{\mathsf{PATH}}$ [32].
5. $\exists\mathcal{A} : \mathsf{EQP}^{\mathcal{A}} \not\subseteq \mathsf{ZPP}^{\mathcal{A}}$ [26].
6. $\exists\mathcal{A} : \mathsf{EQP}^{\mathcal{A}} \not\subseteq \mathsf{NP}^{\mathcal{A}}$ [26].
7. $\exists\mathcal{A} : \mathsf{EQP}^{\mathcal{A}} \not\subseteq \mathsf{P}^{\mathcal{A}}$ [26].
8. $\exists\mathcal{A} : \mathsf{BQEXP}^{\mathcal{A}} \not\subseteq \mathsf{2\text{-}EXP}^{\mathcal{A}}$ [26].
9. $\exists\mathcal{A} : \mathsf{EQP}^{\mathcal{A}} \not\subseteq \mathsf{NP}^{\mathcal{A}} \cup \mathsf{coNP}^{\mathcal{A}}$ [26].
10. $\exists\mathcal{A} : \mathsf{EQP}^{\mathcal{A}} \not\subseteq \mathsf{RP}^{\mathcal{A}}$ [26].
11. $\exists\mathcal{A} : \mathsf{EQP}^{\mathcal{A}} \not\subseteq \mathsf{coRP}^{\mathcal{A}}$ [26].

3.3 Quantum Oracles in a Quantum Setting

1. $\exists\mathcal{A} : \mathsf{QMA}_1^{\mathcal{A}} \not\subseteq \mathsf{QCMA}^{\mathcal{A}}$ [13].
   This implies that $\exists\mathcal{A} : \mathsf{QMA}^{\mathcal{A}} \not\subseteq \mathsf{QCMA}^{\mathcal{A}}$, as $\mathsf{QMA}_1 \subseteq \mathsf{QMA}$.
2. $\exists\mathcal{A} : \mathsf{SQMA}^{\mathcal{A}} \not\subseteq \mathsf{QCMA}^{\mathcal{A}}$ [27].
3. $\exists\mathcal{A} : \mathsf{BQP}^{\mathcal{A}} \not\subseteq \mathsf{QMA}_1^{\mathcal{A}}$ [5].

Chapter 4

Methods and Paradigms

And then there are the endless repetitions, the drudgery, the basic moves practiced over and over again.

—George Leonard,
Mastery (1991)

In this chapter, we present an overview of the methods underlying our results of Chapter 5. In particular, we present the low-level details of some of the literature results, we surveyed in Chapter 3, in order to help us gain valuable insights about the possible solutions to the problems we posed.

4.1 A Classical Oracle in a Classical Setting

We begin by recounting a classical, earth-shattering, result by Baker, Gill, and Solovay [96].

Theorem 4.1. There is some oracle, namely $\mathcal{A}$, such that

$$\mathsf{NP}^{\mathcal{A}} \not\subseteq \mathsf{P}^{\mathcal{A}}. \tag{4.1}$$

That is, there is some oracle $\mathcal{A}$, such that there is some language L which is in the class $\mathsf{NP}^{\mathcal{A}}$, but it is not in the class $\mathsf{P}^{\mathcal{A}}$.

4.1.1 Preliminaries

At first, we point out the equivalence between perceiving oracles as sets and as functions.

Remark 4.2. An oracle $\mathcal{A}$, which, as we earlier saw, is defined as a set, can be viewed as a function, too: the function that emerges when one maps to the number 1 all of the elements of the set at hand, and to the number 0 all of the rest possible alphabet strings. In particular, one has that

$$x \in \mathcal{A}_{\text{set}} \Leftrightarrow \mathcal{A}_{\text{function}}\left(x\right) = 1, \tag{4.2}$$

and

$$x \notin \mathcal{A}_{\text{set}} \Leftrightarrow \mathcal{A}_{\text{function}}\left(x\right) = 0. \tag{4.3}$$

So, from now on, we are going to use $\mathcal{A}_{\text{sets}}$ and $\mathcal{A}_{\text{function}}$, interchangeably, by invoking the common symbol $\mathcal{A}$.

The main method that we will employ, here, is diagonalization. In particular, we will employ a sort of "slow diagonalization" over polynomial-time, and deterministic, Turing machines [80].

Remark 4.3 (Diagonalization)**.** The method of *diagonalization* is a method for proving that a certain object of interest cannot exist. This object is a member of some set A, that participates in some relation $R \subseteq A \times B$, for B being a set, too.

At first, we create a table M that has, as rows, the elements of A, and, as columns, the elements of B. In the case that $(x, y) \in R$, one has that $M_{i,j} = 1$, otherwise $M_{i,j} = 0$, where the number i is the row-number of x, and the number j is the column-number of y. We then create an object D, such that

$$D_{i,i} = \text{something different than } M_{i,i}. \tag{4.4}$$

Now, we observe that the object D cannot emerge either as a row, or a column, of the matrix M, since, for every i, D differs from the i-th row, as well as the i-th column, at its i-th position.

The following example, somehow illustrates the concept of the diagonalization method.

Example 4.4. For the binary matrix

$$M = \left(M_{i,j}\right)_{i,j\in[4]} = \begin{array}{|c|c|c|c|} \hline 1 & 0 & 1 & 0 \\ \hline 1 & 1 & 0 & 1 \\ \hline 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 \\ \hline \end{array}\,, \tag{4.5}$$

we create the 4-tuple

$$\begin{aligned} t &= (t_i)_{i\in[4]} \\ &= (1 - M_{i,i})_{i\in[4]} \\ &= (0,0,1,1)\,. \end{aligned} \tag{4.6}$$

Note that the vector t does not emerge either as a column, or a row, in the matrix M.

We now proceed with the separation.

4.1.2 The Separation

We claim that the language needed, for our separation, is the unary language

$$L = \{1^n \mid \exists x \in \mathcal{A} : |x| = n\}\,. \tag{4.7}$$

Note that this language L encodes, in unary strings, all of the lengths of the members of the oracle $\mathcal{A}$.

Part One: ***Something*** **is in** $\mathsf{NP}^{\mathcal{A}}$

We prove the following lemma.

Lemma 4.5. It is the case that $L \in \mathsf{NP}^{\mathcal{A}}$.

Proof. There is a polynomial non-deterministic algorithm, with access to the oracle $\mathcal{A}$, which establishes that fact: see Algorithm 1. *What does this algorithm do?* The Algorithm 1, on input 1^n, for n being a natural number, guesses an appropriate $x \in \Sigma^n$, and, then, calls the oracle $\mathcal{A}$ to check whether we have $x \in \mathcal{A}$, or not.

Algorithm 1 The procedure that decides L.

```
1: procedure MACHINE_ONE($\Sigma, 1^n, \mathcal{A}$)
Require: $n \in \mathbb{N}$
2:     $x \leftarrow$ MAKE_A_GUESS($\Sigma, n$)
3:     if $\mathcal{A}(x) = 1$ then
4:         return Yes
5:     else
6:         return No
7:     end if
8: end procedure
```

But what is a guess? Guesses are fundamental entities in the non-deterministic computing setting: see Algorithm 2.

Algorithm 2 The procedure that implements guesses.

```
1: procedure MAKE_A_GUESS($\Sigma, n$)
Require: $n \in \mathbb{N}$
2:     $s \leftarrow \epsilon$                    ▷ The symbol $\epsilon$ denotes the empty string.
3:     for $i$ from 1 to $n$ do
4:         guess $\sigma \in \Sigma$      ▷ This is a magic! That is, the ability to select, in
   one step, an appropriate element from the set $\Sigma$.
5:         $s \leftarrow$ CONCATENATION $(s, \sigma)$          ▷ Note that, for two binary
   strings, say, 0101 and 111100, their concatenation is 0101111100.
6:     end for
7:     return $s$
8: end procedure
```

The proof is complete. □

Part Two: *Something* **is not in** $\mathsf{P}^{\mathcal{A}}$

We prove the following lemma.

Lemma 4.6. It is the case that $L \notin \mathsf{P}^{\mathcal{A}}$.

Proof. As we already pointed out, in the Subsection 4.1.1, we are going to employ the method of diagonalization to show that $L \notin \mathsf{P}^{\mathcal{A}}$.

We start by considering an enumeration over all polynomial-time deterministic Turing machines, with oracle $\mathcal{A}$, namely

$$M_1^{\mathcal{A}}, M_2^{\mathcal{A}}, \ldots, M_i^{\mathcal{A}}, \ldots,$$

and, then, we simulate each $M_i^{\mathcal{A}}$, on input 1^i, for at most $i^{\log i}$ steps, in order to fully-define the oracle $\mathcal{A}$ in a way such that every polynomial-time deterministic Turing machine fails to decide L, given that has access to the oracle $\mathcal{A}$. Note that each Turing machine appears infinitely many times in the enumeration, since machines that differ only in unused states, or extra alphabet symbols, are equivalent, in terms of the language that they decide. Throughout the simulations, we mantain a set X, which will keep track of all the strings x such that $x \notin \mathcal{A}$. At the beginning, we have that $X = \emptyset$.

To create the oracle $\mathcal{A}$, we create, at first, a collection of oracles $\{\mathcal{A}_i\}_i$, one for each length value i. At first, we set $\mathcal{A}_0 = \emptyset$. Finally, we have that

$$\mathcal{A} = \bigcup_{i=0}^{\infty} \mathcal{A}_i. \tag{4.8}$$

During our simulations, how do we answer oracle queries of the form $x \overset{?}{\in} \mathcal{A}$? Well, for a given x, in the case that $|x| < i$, say, we use the oracle $\mathcal{A}_i$. That is, we reply with `Yes`, if $\mathcal{A}_i(x) = 1$, and, otherwise, we reply with `No`. In the case where we have $|x| \geq i$, we reply with `No`, and put x into X. We now proceed with the process of creating the oracle $\mathcal{A}$.

1. In the case that $M_i^{\mathcal{A}}(1^i)$ halts and rejects, we want to ensure that $1^i \in L$. To do this, we set

$$\mathcal{A}_i = \mathcal{A}_{i-1} \cup \left\{x \in \{0,1\}^* \mid |x| = i \text{ and } x \notin X\right\}. \tag{4.9}$$

In this way, we make sure that there exists some $x \in \mathcal{A}$, with $|x| = i$. Thus, we get that $1^i \in L$. However, we need to show that

$$\left\{x \in \{0,1\}^* \mid |x| = i \text{ and } x \notin X\right\} \neq \emptyset. \tag{4.10}$$

Indeed, this is the case: The set X contains no more than

$$\sum_{j=1}^{i} j^{\log j} \tag{4.11}$$

elements of length i, since $\sum_{j=1}^{i} j^{\log j}$ is the total number of steps simulated, so far, on all of these oracle machines $M_i^{\mathcal{A}}$. Note that

$$\sum_{j=1}^{i} j^{\log j} < 2^i. \tag{4.12}$$

Thus, there exists an element x, of length equal to i, that is not in X. Wrapping it up, we immediately get that $x \in \mathcal{A}_i \subseteq \mathcal{A}$, therefore $1^i \in L$. Thus, one has that $L\left(M_i^{\mathcal{A}}\right) \neq L$.

2. Now, in the case that $M_i^{\mathcal{A}}\left(1^i\right)$ halts and accepts, we can ensure that $1^i \notin L$ by setting $\mathcal{A}_i = \mathcal{A}_{i-1}$. In this way, there are no strings of length i in $\mathcal{A}$. Yet again, one has that $L\left(M_i^{\mathcal{A}}\right) \neq L$.

3. In the final case, where $M_i^{\mathcal{A}}\left(1^i\right)$ fails to halt in the amount of time allotted, we again set $\mathcal{A}_i = \mathcal{A}_{i-1}$. *But why?* The reason is that, eventually, our Turing machine, at hand, namely $M_i^{\mathcal{A}}$, will emerge again in the enumeration as, say, $M_k^{\mathcal{A}}$, for some natural k, and $M_k^{\mathcal{A}}$ will be such that it halts in the time allotted. Thus, $M_k^{\mathcal{A}}$ is going to get taken care of, according to the previous steps, namely 1 and 2.

The whole procedure, for creating the oracle $\mathcal{A}$, can be found, in procedural form, in the Algorithm 3.

$\square$

The following remark, Remark 4.7, conveys some intuition about the result of the Theorem 4.1.

Remark 4.7. Let

$$\begin{aligned} 1^{\mathbb{N}} &= \left\{1^1, 1^2, \ldots, 1^n, \ldots\right\} \\ &= \left\{1^n\right\}_{n \in \mathbb{N}} \end{aligned} \tag{4.13}$$

be the set that encodes, as unary strings, the natural numbers. Let $\mathcal{M}$ be the set of all Turing machines such that they are deterministic, and of polynomial-time complexity. We devise the table Σ that depicts the elements of the relation $\sigma \subseteq 1^{\mathbb{N}} \times \mathcal{M}^{\mathcal{A}}$, which encodes the pairs $\left(1^i, M_j^{\mathcal{A}}\right)$, such that the machine $M_j^{\mathcal{A}}\left(1^i\right)$ halts and accepts. That is, we get that

$$\Sigma\left(1^i, M_j^{\mathcal{A}}\right) = \begin{cases} M_i^{\mathcal{A}}\left(1^i\right) & \text{if } M_i^{\mathcal{A}}\left(1^i\right) \text{ halts in time } i^{\log i}, \\ ? & \text{otherwise,} \end{cases} \tag{4.14}$$

where we use "?" for the case that $\left(1^i, M_j^{\mathcal{A}}\right)$ fails to halt in the amount of time allotted. For example, the matrix Σ could be of the form

$$\Sigma = \begin{array}{|c||c|c|c|c|c|} \hline & M_1^{\mathcal{A}} & M_2^{\mathcal{A}} & \dots & M_n^{\mathcal{A}} & \dots \\ \hline\hline 1^1 & 1 & 0 & \dots & ? & \dots \\ \hline 1^2 & ? & 0 & \dots & 1 & \dots \\ \hline \vdots & \vdots & \vdots & \ddots & \vdots & \\ \hline 1^n & 0 & 1 & \dots & ? & \dots \\ \hline \vdots & \vdots & \vdots & & \vdots & \ddots \\ \hline \end{array}. \tag{4.15}$$

Now, we create a Turing machine D, with access to the oracle $\mathcal{A}$, such that

$$D^{\mathcal{A}}\left(1^n\right) = \begin{cases} 1 - M_n^{\mathcal{A}}\left(1^n\right) & \text{if } M_i^{\mathcal{A}}\left(1^i\right) \text{ halts in time } i^{\log i}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.16}$$

We are now going to prove that

$$\begin{aligned} L &= \{1^n \mid \exists x \in \mathcal{A} : |x| = n\} \\ &= L\left(D^{\mathcal{A}}\right). \end{aligned} \tag{4.17}$$

For the proof, we will separately prove the involved inclusions $L\left(D^{\mathcal{A}}\right) \subseteq L$, and $L \subseteq L\left(D^{\mathcal{A}}\right)$.

For the first inclusion, let $1^n \in L\left(D^{\mathcal{A}}\right)$. This means that

$$D^{\mathcal{A}}\left(1^n\right) = 1, \tag{4.18}$$

or, equivalently, that

$$M_n^{\mathcal{A}}\left(1^n\right) = 0, \tag{4.19}$$

or, by the above procedure, that

$$\mathcal{A}_n = \mathcal{A}_{n-1} \cup \underbrace{\left\{x \in \{0,1\}^* \mid |x| = i \text{ and } x \notin X\right\}}_{\neq \varnothing}. \tag{4.20}$$

Thus, there is some element of length n in $\mathcal{A}$. Hence, we get that $1^n \in L$.

For the second inclusion, let $1^n \in L$. This means that there is some element of length n in $\mathcal{A}$, or, equivalently, that $\mathcal{A}_n \neq \mathcal{A}_{n-1}$. Thus, one happily has that $M_n^{\mathcal{A}}\left(1^n\right) = 0$. We get that $D^{\mathcal{A}}\left(1^n\right) = 1$, and, so, $1^n \in L\left(D^{\mathcal{A}}\right)$.

Finally, by invoking the diagonalization principle, see the Remark 4.3, we get that the oracle machine $D^{\mathcal{A}}$ cannot emerge as a column of the matrix Σ, of the Equation (4.15), thus, we have that the machine $D^{\mathcal{A}}$ is not a deterministic polynomial-time one. This implies that $L\left(D^{\mathcal{A}}\right) \notin \mathsf{P}^{\mathcal{A}}$, or $L \notin \mathsf{P}^{\mathcal{A}}$.

4.2 A Classical Oracle in a Quantum Setting

Classical oracles are very versatile: they can aso be used in a quantum setting. We prove the following theorem, which can be found in the work by Chen [32], who complements the work by Aaronson [6].

Theorem 4.8 ([32, 6])**.** There is some classical oracle, namely $\mathcal{A}$, such that

$$\mathsf{BQP}^{\mathcal{A}} \not\subseteq \mathsf{BPP}^{\mathcal{A}}_{\mathsf{PATH}}. \tag{4.21}$$

That is, there is some oracle $\mathcal{A}$, such that there is some language L which is in the class $\mathsf{BQP}^{\mathcal{A}}$, but is not in the class $\mathsf{BPP}^{\mathcal{A}}_{\mathsf{PATH}}$.

4.2.1 Preliminaries

We start by introducing some basic machinery, in the following definitions.

Definition 4.9 (Various Definitions)**.** Let $M \in \mathbb{N}$, and let

$$z = z_1 z_2 \dots z_M \in \{0,1\}^M$$

be a binary string. Then, a *literal* is of the form z_i, or $1 - z_i$. A *k-term* is a product of k literals, each involving a different z_i. This product is 1 if the literals take on their prescribed values, and 0, otherwise. Finally, we denote by $\mathcal{U}$ the uniform distribution over $\{0,1\}^M$.

Definition 4.10. A distribution $\mathcal{D}$, over $\{0,1\}^M$, is ε-almost k-wise equivalent to $\mathcal{U}$, if, for every k-term C, we have that

$$1 - \varepsilon \leq \frac{\Pr_{\mathcal{D}}[C]}{\Pr_{\mathcal{U}}[C]} \leq 1 + \varepsilon. \tag{4.22}$$

Note 4.11. Note that

$$\Pr_{\mathcal{U}}[C] = \frac{1}{2^k}, \tag{4.23}$$

since $\mathcal{U}$ is the uniform distribution over $\{0,1\}^M$, and C consists of k literals.

Definition 4.12. Given two distributions, namely $\mathcal{D}_1$ and $\mathcal{D}_2$, over $\{0,1\}^M$, we say that $\mathcal{D}_1$ ε-almost k-wise dominates $\mathcal{D}_2$, if, for every k-term C, one has that

$$\frac{\Pr_{\mathcal{D}_1}[C]}{\Pr_{\mathcal{D}_2}[C]} \geq 1 - \varepsilon. \tag{4.24}$$

Also, we say that $\mathcal{D}_1$ and $\mathcal{D}_2$ are ε-almost k-wise equivalent if they ε-almost k-wise dominate each other, that is, if, for every k-term C, we have that

$$1 - \varepsilon \leq \frac{\Pr_{\mathcal{D}_1}[C]}{\Pr_{\mathcal{D}_2}[C]} \leq 1 + \varepsilon. \tag{4.25}$$

4.2.2 The Separation

The problem, that we are going to use for our separation, draws upon the Fourier transform and the concept of statistical correlation, that is, it is named `FORELLATION`, and is described in the Table 4.1.

We are now going to introduce, and, in part, remind to the reader, some very useful input distributions, namely $\mathcal{F}$, $\mathcal{F}'$, $\mathcal{U}$, and $\mathcal{U}'$.

Definition 4.13. A sample $\langle f, g \rangle$ from the distribution $\mathcal{F}$ is generated as follows. First, choose a random real vector

$$v = (v_x)_{x \in \{0,1\}^n} \in \mathbb{R}^N = \mathbb{R}^{2^n},$$

by drawing each entry independently from a Gaussian distribution with mean value 0 and variance 1. Then, set

$$f(x) = \operatorname{sgn}(v_x), \tag{4.34}$$

and

$$g(x) = \operatorname{sgn}(\hat{v}_x), \tag{4.35}$$

for all x. Here,

$$\operatorname{sgn}(\alpha) = \begin{cases} 1 & \text{if } \alpha \geq 0, \\ -1 & \text{otherwise,} \end{cases} \tag{4.36}$$

and $\hat{v}_y$ is the Fourier transform of v_y, over $\mathbb{Z}_2^n$, that is,

$$\hat{v}_y = \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} (-1)^{xy} v_x. \tag{4.37}$$

In other words, the functions f and g individually are still uniformly random, but they are no longer independent: now g is extremely correlated with the Fourier transform of f. Hence, "forrelated."

Table 4.1: The problem FORRELATION.

FORELLATION ($\texttt{FOR}_n$)

Input Two Boolean functions

$$f : \{0,1\}^n \rightarrow \{-1,1\}, \tag{4.26}$$

and

$$g : \{0,1\}^n \rightarrow \{-1,1\}, \tag{4.27}$$

that map binary strings, of length n, to the set $\{-1,1\}$.

Output For

$$\Phi_{f,g} = \frac{1}{2^{3n/2}} \sum_{x,y \in \{0,1\}^n} f(x)(-1)^{xy} g(y) \tag{4.28}$$

decide whether it is the case that

$$\left|\Phi_{f,g}\right| \leq 0.01, \tag{4.29}$$

that is, the Yes case, or the case that

$$\left|\Phi_{f,g}\right| \geq 0.07, \tag{4.30}$$

that is, the No case. We note that

$$\left|\Phi_{f,g}\right| \leq 1. \tag{4.31}$$

We will use $\texttt{FOR}_n$ to denote the FORRELATION problem, when parameterized by n. That is,

$$\texttt{FOR}_n(f,g) = 1 \Leftrightarrow \left|\Phi_{f,g}\right| \leq 0.01, \tag{4.32}$$

and

$$\texttt{FOR}_n(f,g) = 0 \Leftrightarrow \left|\Phi_{f,g}\right| \geq 0.07. \tag{4.33}$$

Definition 4.14. The distribution $\mathcal{F}'$, is the conditional distribution obtained by $\mathcal{F}$ conditioned on the event that

$$\left|\Phi_{f,g}\right| \geq 0.07. \tag{4.38}$$

That is, a sample $\langle f,g \rangle$ from $\mathcal{F}'$ can be generated as follows: We draw a sample $\langle f,g \rangle$ from $\mathcal{F}$, and compute $\left|\Phi_{f,g}\right|$. If $\left|\Phi_{f,g}\right| \geq 0.07$, then we return $\langle f,g \rangle$, otherwise we discard $\langle f,g \rangle$, and repeat the experiment until the terminating condition is met. In a similar way, we obtain the distribution $\mathcal{U}'$, by the distribution $\mathcal{U}$, conditioned on the event that

$$\left|\Phi_{f,g}\right| \leq 0.01. \tag{4.39}$$

Lemma 4.15. For any $k = N^{o(1)}$, the distributions $\mathcal{F}'$ and $\mathcal{U}'$ are $o(1)$-almost k-wise independent.

Theorem 4.16. For a subset $D \subseteq \{0,1\}^M$, fix a partial function

$$f : D \to \{0,1\}. \tag{4.40}$$

Suppose that there are two distributions, namely $\mathcal{D}_0$ and $\mathcal{D}_1$, supported on the 0-inputs and the 1-inputs, respectively, such that they are $o(1)$-almost k-wise equivalent. Then, there are no $\mathsf{BPP}_{\mathsf{PATH}}$-machines that can compute the function f using at most k queries.

Proof. Let M be a $\mathsf{BPP}^{\mathcal{A}}_{\mathsf{PATH}}$-machine that computes f. Then, let the probability function $a(x)$ be the success probability of M, that is, the probability of something being postselected successfully, and let the probability function $s(x)$ be the successfully-accepting probability of M, that is, the probability of something being both postselected successfully, and the relevant computation being accepting, respectively, on the input x. For some distribution, namely $\mathcal{D}$, over $\{0,1\}^M$, let

$$a(\mathcal{D}) = \mathbb{E}_{x\sim\mathcal{D}}\left[a(x)\right], \tag{4.41}$$

and

$$s(\mathcal{D}) = \mathbb{E}_{x\sim\mathcal{D}}\left[s(x)\right]. \tag{4.42}$$

By the definition of the $\mathsf{BPP}_{\mathsf{PATH}}$-machines, and the fact that $\mathcal{D}_0$ is supported on 0-inputs, and $\mathcal{D}_1$ is supported on 1-inputs, we have that

$$a(\mathcal{D}_0) \leq \frac{1}{3}s(\mathcal{D}_0). \tag{4.43}$$

and

$$a(\mathcal{D}_1) \geq \frac{2}{3}s(\mathcal{D}_1), \tag{4.44}$$

Since M makes at most k queries, the probability $a(x)$ can be written as

$$a(x) = \sum_{i=1}^{m} a_i C_i(x), \tag{4.45}$$

where $a_i \geq 0$, $m \in \mathbb{N}$, and each C_i is a k'-term, for $k' \leq k$, for every i. Therefore, by using the fact that $\mathcal{D}_0$ and $\mathcal{D}_1$ are $o(1)$-almost k-wise equivalent, we have that

$$\begin{aligned} a(\mathcal{D}_1) &= \sum_{i=1}^{m} \mathbb{E}_{x\sim\mathcal{D}_1}[a_i C_i] \\ &\geq (1-o(1)) \sum_{i=1}^{m} \mathbb{E}_{x\sim\mathcal{D}_0}[a_i C_i] \\ &= (1-o(1))\, a(\mathcal{D}_0). \end{aligned} \tag{4.46}$$

Similarly, we have that

$$a(\mathcal{D}_0) \geq (1-o(1))\, a(\mathcal{D}_1). \tag{4.47}$$

Hence,

$$1 - o(1) \leq \frac{a(\mathcal{D}_0)}{a(\mathcal{D}_1)} \leq 1 + o(1), \tag{4.48}$$

and, by a same, as above, argument,

$$1 - o(1) \leq \frac{a(\mathcal{D}_0)}{a(\mathcal{D}_1)} \leq 1 + o(1). \tag{4.49}$$

Thus,

$$1 - o(1) \leq \frac{a(\mathcal{D}_1)/s(\mathcal{D}_1)}{a(\mathcal{D}_0)/s(\mathcal{D}_0)} \leq 1 + o(1), \tag{4.50}$$

which contradicts the facts embodied in the Equations (4.43) and (4.44). This completes the proof. □

Part One: ***Something*** **is in** $\mathsf{BQP}^{\mathcal{A}}$

We prove the following lemma.

Lemma 4.17. It is the case that $\mathtt{FOR_n} \in \mathsf{BQP}^{\mathcal{A}}$.

Proof. To show $\mathtt{FOR_n} \in \mathsf{BQP}^{\mathcal{A}}$ we just need to present a $\mathsf{BQP}^{\mathcal{A}}$-machine that solves $\mathtt{FOR_n}$.

We start with the initial state

$$\begin{aligned}
|\psi_0\rangle &= \bigotimes_{i=1}^{n} |0\rangle \\
&= \underbrace{|0\rangle \otimes |0\rangle \otimes \cdots \otimes |0\rangle}_{n} \\
&= |0\rangle^{\otimes n} , \qquad (4.51)
\end{aligned}$$

and, then, we apply a composite Hadamard transformation $H^{\otimes n}$, to the vector $|\psi_0\rangle$, to get

$$\begin{aligned}
|\psi_1\rangle &= H^{\otimes n} |\psi_0\rangle \\
&= H^{\otimes n} |0\rangle^{\otimes n} \\
&= \underbrace{H|0\rangle \otimes H|0\rangle \otimes \cdots \otimes H|0\rangle}_{n} \\
&= \left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right)^{\otimes n} \\
&= \left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) \otimes \left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) \otimes \cdots \\
&\quad \cdots \otimes \left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) \\
&= \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \\
&= \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle . \qquad (4.52)
\end{aligned}$$

Then, we query f, in superposition, to get

$$|\psi_2\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} f(x) |x\rangle , \qquad (4.53)$$

or, after we apply a composite Hadamard transformation $H^{\otimes n}$,

$$|\psi_3\rangle = \frac{1}{N} \sum_{x \in \{0,1\}^n} \sum_{y \in \{0,1\}^n} f(x) (-1)^{xy} |y\rangle , \qquad (4.54)$$

or, after we query g, in superposition, by invoking the oracle $\mathcal{A}$,

$$|\psi_4\rangle = \frac{1}{N} \sum_{x \in \{0,1\}^n} \sum_{y \in \{0,1\}^n} f(x) (-1)^{xy} g(y) |y\rangle , \qquad (4.55)$$

or, after we apply a composite Hadamard transformation $H^{\otimes n}$,

$$\begin{aligned}|\psi_5\rangle &= \frac{1}{N^{3/2}} \sum_{x\in\{0,1\}^n} \sum_{y\in\{0,1\}^n} \sum_{z\in\{0,1\}^n} f(x)(-1)^{xy} g(y)(-1)^{yz} |y\rangle \\ &= \frac{1}{N^{3/2}} \sum_{s\in\{0,1\}^n} \alpha_s |s\rangle . \end{aligned} \tag{4.56}$$

Suppose, now, that we measure $|\psi_5\rangle$, in the computational basis. Let the symbol M denote the random variable which denotes the label of the resulting, after the measurement, basis state. We now observe that

$$\begin{aligned}\Pr[M = 00\ldots0] &= |\langle 00\ldots0|\psi_5\rangle|^2 \\ &= |\alpha_{00\ldots0}|^2 \\ &= \left| \frac{1}{N^{3/2}} \sum_{x\in\{0,1\}^n} \sum_{y\in\{0,1\}^n} f(x)(-1)^{xy} g(y) \right|^2 \\ &= \left(\frac{1}{N^{3/2}} \sum_{x\in\{0,1\}^n} \sum_{y\in\{0,1\}^n} f(x)(-1)^{xy} g(y) \right)^2 \\ &= \frac{1}{N^3} \left(\sum_{x\in\{0,1\}^n} \sum_{y\in\{0,1\}^n} f(x)(-1)^{xy} g(y) \right)^2 \\ &= \frac{1}{2^{3n}} \left(\sum_{x\in\{0,1\}^n} \sum_{y\in\{0,1\}^n} f(x)(-1)^{xy} g(y) \right)^2 \\ &= \frac{1}{2^{3n}} \left(\sum_{x,y\in\{0,1\}^n} f(x)(-1)^{xy} g(y) \right)^2 \\ &= \Phi^2_{f,g}. \end{aligned} \tag{4.57}$$

Equation (4.57) implies that, in order to compute $|\Phi_{f,g}| = \sqrt{\Phi^2_{f,g}}$, we need to somehow estimate the probability of observing $00\ldots0$, after performing some measurement on the entire system. So, we measure our whole system, in the computational basis, polynomially many times. Before each measurement, we re-prepare it—as measurements destroy the quantumness of a quantum system. $\square$

Part Two: *Something* **is not in** $\mathsf{BPP}^{\mathcal{A}}_{\mathsf{PATH}}$

We prove the following lemma.

Lemma 4.18. It is the case that $\texttt{FOR}_\texttt{n} \notin \mathsf{BPP}^{\mathcal{A}}_{\mathsf{PATH}}$.

Proof. We are going to prove that there are no $\mathsf{BPP}_{\mathsf{PATH}}$-machines for deciding the problem FORRELATION, when allotted polynomially many queries.

By the definition of $\mathcal{F}'$ and $\mathcal{U}'$, and Lemma 4.15, we can see that $\mathcal{F}'$ is supported on the 1-inputs, of $\texttt{FOR}_n$, and, similarly, $\mathcal{U}'$ is supported on the 0-inputs, of $\texttt{FOR}_n$. Also, they are both $o(1)$-almost $N^{o(1)}$-wise independent. This means that they are also $o(1)$-almost $N^{o(1)}$-wise equivalent. Since we have that $N^{o(1)} \in \text{poly}(n)$, the result follows from Theorem 4.16. Finally, by using a standard diagonalization argument, one can show that FORRELATION is not in the class $\mathsf{BPP}^{\mathcal{A}}_{\mathsf{PATH}}$, and the proof is complete. □

4.3 A Quantum Oracle in a Quantum Setting

We state, and prove, the following theorem, namely Theorem 4.19. The Theorem 4.19, and its proof, were drawn from the relevant work [5] by Aaronson. However, the first, and original, example of an oracle separation that uses quantum oracles, is due to some other paper [13] by Aaronson and Kuperberg.

Theorem 4.19. There is some quantum oracle, namely $\mathcal{A}$, such that

$$\mathsf{QMA}^{\mathcal{A}} \not\subseteq \mathsf{QMA}_1^{\mathcal{A}}. \tag{4.58}$$

That is, there is some oracle $\mathcal{A}$, such that there is some language L which is in the class $\mathsf{QMA}^{\mathcal{A}}$, but it is not in the class $\mathsf{QMA}_1^{\mathcal{A}}$.

4.3.1 Preliminaries

We will draw some lemmata from the relevant work by Aaronson [5]. All of the proofs, to these lemmata, can, of course, be found in the afore-mentioned work by Aaronson [5].

Lemma 4.20 ([5])**.** Let

$$p(\theta)(x) = b_0(\theta) + b_1(\theta)x + b_2(\theta)x^2 + \cdots + b_N(\theta)x^N \tag{4.59}$$

be a real polynomial, on the variable x, with all-real roots, parameterized by the quantity $\theta \in \mathbb{R}$. Suppose that the coefficients

$$b_0(\theta), b_1(\theta), \ldots, b_N(\theta)$$

are all analytic functions on θ. Then, there exist real analytic functions

$$\lambda_0(\theta), \lambda_1(\theta), \ldots, \lambda_N(\theta)$$

such that

$$\{\lambda_0(\theta), \lambda_1(\theta), \ldots, \lambda_N(\theta)\}$$

is the set of all the roots of the polynomial $p(\theta)(x)$, for all $\theta \in \mathbb{R}$.

Lemma 4.21 ([5])**.** Let $f : \mathbb{R} \to \mathbb{R}$ be a real analytic function. If there exists some open interval $(x, y) \subsetneq \mathbb{R}$, for real numbers x and y, on which f is constant, then f is constant everywhere.

Lemma 4.22 ([5])**.** Let V be a quantum verifier that takes as input a quantum witness on Q qubits, namely $|\phi\rangle$, and that makes T queries to a quantum oracle, that is described by a unitary matrix U. Also, let $a(U)$ be the acceptance probability of V^U, maximized over all possible witnesses $|\phi\rangle$. Then, there exists a $2^Q \times 2^Q$ matrix, namely $E(U)$, whose entries are all complex numbers, and is such that

1. every entry of the matrix $E(U)$ is polynomial in the entries of U, of degree at most $2T$,
2. the matrix $E(U)$ is Hermitean, for all U, and
3. the acceptance probability $a(U)$ equals the largest eigenvalue of the matrix $E(U)$, for all unitaries U.

We now proceed with the separation.

4.3.2 The Separation

We define the transformation

$$U_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \tag{4.60}$$

which is just a rotating function on θ. Now, using this, we can define a family of oracles

$$\mathcal{A} = \{U_{\theta_n} \mid \theta_n \in [1, 2] \cup \{0\}\}$$

$$\begin{aligned}
&= \{U_{\theta_n}\}_{\theta_n \in [1,2] \cup \{0\}} \\
&= \{U_{\theta_n}\}_{\theta_n \in [1,2]} \cup \{U_{\theta_n}\}_{\theta_n \in \{0\}} \\
&= \{U_{\theta_n}\}_{\theta_n \in [1,2]} \uplus \{U_{\theta_n}\}_{\theta_n \in \{0\}} \\
&= \mathcal{A}_{\text{good}} \uplus \mathcal{A}_{\text{bad}}. \qquad (4.61)
\end{aligned}$$

To this point, we can encode the "good" oracles of the set

$$\mathcal{A}_{\text{good}} = \{U_{\theta_n}\}_{\theta_n \in [1,2]}, \qquad (4.62)$$

to form the unary language

$$\begin{aligned}
L &= \left\{1^n \mid U_{\theta_n} \in \mathcal{A}_{\text{good}}\right\} \\
&= \{1^n \mid \theta_n \text{ corresponds to some good } U_{\theta_n}\} \\
&= \{1^n \mid \theta_n \in [1,2]\}. \qquad (4.63)
\end{aligned}$$

By using this language L, and the oracle $\mathcal{A}$, we are going to prove our separation.

Part One: ***Something*** **is in** $\mathsf{QMA}^{\mathcal{A}}$

We prove the following lemma.

Lemma 4.23. It is the case that $L \in \mathsf{QMA}^{\mathcal{A}}$.

Proof. We are going to prove something stronger, that is, that $L \in \mathsf{BQP}^{\mathcal{A}}$. Since $\mathsf{BQP} \subseteq \mathsf{QMA}$, for every world, that is, even for relativized ones, the desired result follows.

The transformation

$$U_\theta = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, \qquad (4.64)$$

can be carried out in quantum polynomial time, with bounded error. *Why so?* Note that what we only have to do is to prepare the basis ket

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

and, then, apply U_θ, to $|0\rangle$, to get

$$|\psi\rangle = U_\theta |0\rangle$$

$$\begin{aligned}
&= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} \\
&= \cos\theta \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \sin\theta \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\
&= \cos\theta \left|0\right\rangle + \sin\theta \left|1\right\rangle . \qquad (4.65)
\end{aligned}$$

We then measure in the computational basis, and we perceive the outcome as a random variable M. The probability of getting "1," as a measurement-outcome, is

$$\begin{aligned}
\Pr\left[M = 1\right] &= \left|\langle\psi|1\rangle\right|^2 \\
&= \left|(\cos\theta \left\langle 0\right| + \sin\theta \left\langle 1\right|) \left|1\right\rangle\right|^2 \\
&= \left|\cos\theta \left\langle 0\right| \left|0\right\rangle + \sin\theta \left\langle 1\right| \left|1\right\rangle\right|^2 \\
&= \left|\cos\theta \left\langle 0|1\right\rangle + \sin\theta \left\langle 1|1\right\rangle\right|^2 \\
&= \left|\cos\theta \cdot 0 + \sin\theta \cdot 1\right|^2 \\
&= \sin^2\theta. \qquad (4.66)
\end{aligned}$$

In the `Yes` case, that is, in the case where $\theta \in [1, 2]$, we get that

$$\begin{aligned}
\Pr\left[M = 1\right] &= \sin^2\theta \\
&= 0 + \varepsilon, \qquad (4.67)
\end{aligned}$$

for $\varepsilon > 0$, and, in the `No` case, where $\theta = 0$, we get that

$$\begin{aligned}
\Pr\left[M = 1\right] &= \sin^2(0) \\
&= 0. \qquad (4.68)
\end{aligned}$$

What we observe, here, is that there is an ε-gap between the acceptance probabilities in the `Yes` and the `No` cases. Thus, we can amplify this, by using one of the standard amplifying procedures [80, 103], and get a BQP acceptance schema. We now can conclude that $L \in \mathsf{BQP}^{\mathcal{A}}$. □

Part Two: *Something* **is not in** $\mathsf{QMA}_1^{\mathcal{A}}$

We prove the following lemma.

Lemma 4.24 ([5])**.** For

$$\begin{aligned} L &= \left\{1^n \mid U_{\theta_n} \in \mathcal{A}_{\text{good}}\right\} \\ &= \{1^n \mid \theta_n \text{ corresponds to some good } U_{\theta_n}\} \\ &= \{1^n \mid \theta_n \in [1,2]\}, \end{aligned} \tag{4.69}$$

it is the case that $L \notin \mathsf{QMA}_1^{\mathcal{A}}$.

Proof. Let V be a quantum verifier, let T be the number of queries that the verifier V makes to U, and let Q be the number of qubits of the quantum witness, or proof, of the verifier V. Let $a(U)$ be the acceptance probability of V, assuming $U = U_\theta$, maximized over all Q-qubit witnesses $|\psi\rangle$. By invoking Lemma 4.22, we get a $2^Q \times 2^Q$ matrix $E(U)$, with complex entries, that satisfies the three properties of Lemma 4.22. Now, let $N = 2^Q$, and let

$$\lambda_0(\theta), \lambda_1(\theta), \ldots, \lambda_N(\theta)$$

be the eigenvalues of $E(U)$. Then, these $\lambda_i(\theta)$ are the roots of some polynomial $p(\theta)(x)$, of degree N, namely

$$p(\theta)(x) = b_0(\theta) + b_1(\theta)x + b_2(\theta)x^2 + \cdots + b_N(\theta)x^N. \tag{4.70}$$

Each of the coefficients $b_i(\theta)$ is a polynomial in the entries of $E(\theta)$, of degree at most N, and, hence, by the property 1, each of these coefficients is a polynomial, in $\cos\theta$ and $\sin\theta$, of degree at most $2TN$. By the property 2, all of the eigenvalues $\lambda_i(\theta)$ are real, and, thus, the coefficients $b_j(\theta)$ must be all real, as well, for every θ. Thus, each $b_j(\theta)$ is a real analytic function on θ. By Lemma 4.20, we can take every $\lambda_i(\theta)$ to be a real analytic function. By the property 3, we have that

$$a(\theta) = \max_{i\in\{0,1,\ldots,N\}} \lambda_i(\theta). \tag{4.71}$$

If V is a valid QMA_1 verifier, then we must have

$$a(0) \leq \frac{1}{2}, \tag{4.72}$$

and

$$\forall\theta \in [1,2] : a(\theta) = 1. \tag{4.73}$$

Since N is finite, there exists some $i \in [N]$ such that

$$\lambda_i(0) \leq \frac{1}{2}, \tag{4.74}$$

and

$$\forall \theta \in (x, y) \subsetneq [1, 2] : \lambda_i(\theta) = 1. \tag{4.75}$$

However, this contradicts the analyticity of λ_i, by the Lemma 4.21. Thus, there is a choice for θ such that V does not solve the problem correctly, when given the unitary U_θ as oracle.

We now diagonalize, over all naturals n, to achieve the desired oracle separation. We claim that $L \notin \mathsf{QMA}_1^{\mathcal{A}}$. Let

$$M_1, M_2, \ldots, M_i, \ldots$$

be an enumeration of all QMA_1-machines. Then, for each i, we choose n_i to be so large, such that U_{n_i} cannot have been queried by any of the machines

$$M_1, M_2, \ldots, M_{i-1}.$$

Finally, we set θ_{n_i} so that M_i fails on input 1^{n_i}. Now, we see that either we have $\theta_n = 0$, and there exists a witness $|\psi\rangle$ causing M_i to accept with probability greater than $1/2$, or, else, we have that $\theta_n \in [1, 2]$, and no witness causes M_i to accept with probability 1. Hence, it is impossible for any of all the QMA_1-machines to decide L. $\square$

Algorithm 3 The procedure that fully defines the classical oracle $\mathcal{A}$, used for the separation of NP from P.

```
1: procedure MAKE_ORACLE
2:     $X \leftarrow \emptyset$
3:     $\mathcal{A}_0 \leftarrow \emptyset$
4:     for all $i \in \mathbb{N} = \{1, 2, \dots\}$ do
5:         $j \leftarrow 1$
6:         simulate the first step of $M_i^{\mathcal{A}_{i-1}}(1^i)$
7:         while $M_i^{\mathcal{A}_{i-1}}(1^i)$ has not halted and $j \leq i^{\log i}$ do
8:             if the $j$-th step of $M_i^{\mathcal{A}_{i-1}}(1^i)$ involves some oracle call then
9:                 if $|x| < i$ then
10:                    if $x \in \mathcal{A}_{i-1}$ then
11:                        answer "Yes" to the oracle call
12:                    else
13:                        answer "No" to the oracle call
14:                    end if
15:                else
16:                    answer "No" to the oracle call    ▷ Answer "No" to the unknown string.
17:                    $X \leftarrow X \cup \{x\}$
18:                end if
19:            else
20:                simulate the $j$-th step of $M_i^{\mathcal{A}_{i-1}}(1^i)$
21:            end if
22:            $j \leftarrow j + 1$
23:        end while
24:        if $M_i^{\mathcal{A}_{i-1}}(1^i)$ halted then
25:            if $M_i^{\mathcal{A}_{i-1}}(1^i)$ rejected then
26:                $\mathcal{S} \leftarrow \{x \in \{0,1\}^* \mid |x| = i \text{ and } x \notin X\}$
27:                $\mathcal{A}_i \leftarrow \mathcal{A}_{i-1} \cup \mathcal{S}$    ▷ Note that the set $\mathcal{S}$ is always non-empty.
28:            else
29:                $\mathcal{A}_i \leftarrow \mathcal{A}_{i-1}$
30:            end if
31:        else
32:            $\mathcal{A}_i \leftarrow \mathcal{A}_{i-1}$
33:        end if
34:    end for
35:    $\mathcal{A} \leftarrow \bigcup_{i=1}^{\infty} \mathcal{A}_i$
36:    return $\mathcal{A}$
37: end procedure
```

Chapter 5

Results

Every now and then I get the feeling I could do a good work. Yet what have I done. What I have done, nonetheless, is quite good, some of it, and with work I should do better. One indication: one story accepted.

—Sylvia Plath,
The Unabridged Journals of Sylvia Plath, 1950–1962 (2000)

In this chapter, we present our main, and in some sense novel, result. In particular, we proved that there exists some quantum oracle $\mathcal{A}$, such that the class $\mathsf{SQMA}_1^{\mathcal{A}}$ is not a subset of the class $\mathsf{QCMA}^{\mathcal{A}}$, or

$$\exists \mathcal{A} \in \textsc{oracles} : \mathsf{SQMA}_1^{\mathcal{A}} \not\subseteq \mathsf{QCMA}^{\mathcal{A}}, \tag{5.1}$$

where "$\textsc{oracles}$" denotes the set of all oracles.

5.1 Oracle Separation of SQMA_1 from QCMA

As it turns out, there is a relativized world, induced by calls to some quantum oracle $\mathcal{A}$, in which QCMA is unable to capture the computational power of the subset-state version of QMA, namely SQMA, even when there exists a perfect subset-state-flavored proof out there.

Note 5.1. In many parts, we follow the recent seminal work by Fefferman and Kimmel [27]. Note that they call their main oracle structure "preimage-correct," while we use the related term "preimage-appropriate," to name our own main oracle structures.

5.1.1 Preliminaries

Let $2n$, for $n \in \mathbb{N}$ denote the length of the members, namely x, of exponentially large, in n, sets S of some class of sets $\mathbf{S}$. That is, $x \in S \in \mathbf{S}$. We want to examine some, not obvious, properties of these sets S. We have that

$$N = 2^n = |S| \,. \tag{5.2}$$

We now give some more definitions, in the Definitions 5.2, 5.4, and 5.6.

Definition 5.2. Following Fefferman and Kimmel [27], we define

$$\boldsymbol{\sigma}^n = \text{the set of permutations } \sigma \text{ from } \left[N^2\right] \text{ to } \left[N^2\right], \tag{5.3}$$

$$S_{\text{pre}}\left(\sigma\right) = \left\{j \mid \sigma\left(j\right) \in \left[N\right]\right\}, \tag{5.4}$$

$$\boldsymbol{\sigma}_{\text{pre}}\left(S\right) = \left\{\sigma \mid \sigma \in \boldsymbol{\sigma}^n \text{ and } S_{\text{pre}}\left(\sigma\right) = S\right\}, \tag{5.5}$$

$$\mathbf{S}^n_{\text{odd}} = \left\{S \mid S \subseteq \left[N^2\right] \text{ and } |S| = N \text{ and } |S \cap \mathbb{Z}_{\text{odd}}| = N\right\}, \tag{5.6}$$

and

$$\mathbf{S}^n_{\text{bal}} = \left\{S \mid S \subseteq \left[N^2\right] \text{ and } |S| = N \text{ and } |S \cap \mathbb{Z}_{\text{odd}}| = N/2\right\}. \tag{5.7}$$

Remark 5.3. Note that "odd" stands for "all-odd," and that "bal" stands for "balanced." The set $\mathbf{S}^n_{\text{odd}}$ contains all the subsets S of $\left[N^2\right]$, of size N, that contain only odd elements, and the set $\mathbf{S}^n_{\text{bal}}$ contains all the subsets S of $\left[N^2\right]$, of size N, that contain half-odd, and half-even, elements.

We are going to use those classes of sets, to draw sets that constitute the preimages of permutations. These preimages, here, are about the first N elements of $\left[N^2\right]$, or, equivalently, the elements of $[N]$. For every one of the aforementioned permutations $\sigma \in \boldsymbol{\sigma}^n$, one has that

$$\sigma : S \to [N] \subseteq \left[N^2\right], \tag{5.8}$$

where $S \in \mathbf{S}^n_{\text{odd}} \cup \mathbf{S}^n_{\text{bal}}$.

Definition 5.4. Let $\boldsymbol{\sigma} \subseteq \boldsymbol{\sigma}^n$. Then, if ρ is a density matrix, and $\mathcal{P}_\sigma$ is a unitary that applies the in-place permutation σ, that is,

$$\mathcal{P}_\sigma \left|i\right\rangle = \left|\sigma\left(i\right)\right\rangle, \tag{5.9}$$

then we have that

$$\mathscr{P}_{\boldsymbol{\sigma}}\left(\rho\right) = \frac{1}{|\boldsymbol{\sigma}|} \sum_{\sigma \in \boldsymbol{\sigma}} \mathcal{P}_\sigma \rho \mathcal{P}_\sigma^\dagger, \tag{5.10}$$

is a CPTP map, with

$$\frac{1}{|\boldsymbol{\sigma}|}\sum_{\sigma\in\boldsymbol{\sigma}}\mathcal{P}_\sigma^\dagger\mathcal{P}_\sigma = I. \tag{5.11}$$

Note 5.5. In many cases we are going to write $\mathscr{P}_{\boldsymbol{\sigma}}\left(|\psi\rangle\right) = \mathscr{P}_{\boldsymbol{\sigma}}|\psi\rangle$, but what we will mean is $\mathcal{P}_\sigma|\psi\rangle$, for some permutation $\sigma\in\boldsymbol{\sigma}$. That is, we pick a, uniformly at random, permutation to apply to $|\psi\rangle$.

In the Definition 5.6, we introduce the first important class of oracles that we are going to use in this paper.

Definition 5.6 (Randomized-preimage-appropriate Oracles)**.** Let $\mathcal{A}$ be a countably-infinite set of quantum operators, that is, of Completely-Positive Trace-Preserving (CPTP) maps, namely

$$\mathcal{A} = \{\mathscr{A}_1, \mathscr{A}_2, \dots\}, \tag{5.12}$$

where each $\mathscr{A}_n$ implements an operation on $2n$ qubits. We say that $\mathcal{A}$ is a randomized-preimage-appropriate oracle, if, for every natural n, we have that

$$\mathscr{A}_n = \mathscr{P}_{\boldsymbol{\sigma}_{\text{pre}}(S)}, \tag{5.13}$$

for some $S\in\mathbf{S}^n_{\text{odd}}\cup\mathbf{S}^n_{\text{bal}}$.

Finally, we state a useful lemma, Lemma 5.7, whose proof can be found in the work by Fefferman and Kimmel [27].

Lemma 5.7 (Adversary Bound for Permutation Oracles)**.** Let $\boldsymbol{\sigma}\subseteq([V]\to[V])$ be a set of permutations acting on the elements of the set $[V]$. Let $f:\boldsymbol{\sigma}\to\{0,1\}$ be a function on permutations. Let $\boldsymbol{\sigma}_X\subseteq\boldsymbol{\sigma}$ be a set such that if $\sigma\in\boldsymbol{\sigma}_X$, then $f(\sigma)=1$. Let $\boldsymbol{\sigma}_Y\subseteq\boldsymbol{\sigma}$ be a set that if $\sigma\in\boldsymbol{\sigma}_Y$, then $f(\sigma)=0$. Let $R\subseteq\boldsymbol{\sigma}_X\times\boldsymbol{\sigma}_Y$ be a relation such that

- for every $\sigma_x\in\boldsymbol{\sigma}_X$, there exist at least m different permutations $\sigma_y\in\boldsymbol{\sigma}_Y$, such that $\left(\sigma_x,\sigma_y\right)\in R$.

- For every $\sigma_y\in\boldsymbol{\sigma}_Y$, there exist at least m' different permutations $\sigma_x\in\boldsymbol{\sigma}_X$, such that $\left(\sigma_x,\sigma_y\right)\in R$.

- Let $l_{x,i}$ be the number of $\sigma_y \in \boldsymbol{\sigma}_Y$ such that $(\sigma_x, \sigma_y) \in R$, and $\sigma_x(i) \neq \sigma_y(i)$. Let $l_{y,i}$ be the number of $\sigma_x \in \boldsymbol{\sigma}_X$ such that $(\sigma_x, \sigma_y) \in R$, and $\sigma_x(i) \neq \sigma_y(i)$. Let, also,

$$l_{max} = \max_{(\sigma_x,\sigma_y)\in R,\, i} l_{x,i} l_{y,i}. \tag{5.14}$$

Then, given an in-place permutation oracle $\mathcal{P}_\sigma$, for $\sigma \in \boldsymbol{\sigma}$, that acts as

$$\mathcal{P}_\sigma |i\rangle = |\sigma(i)\rangle, \tag{5.15}$$

any quantum algorithm that correctly evaluates $f(\sigma)$, with probability at least $1 - \varepsilon$, for every element σ of $\boldsymbol{\sigma}_X$, or $\boldsymbol{\sigma}_Y$, must use at least

$$\left(1 - 2\sqrt{\varepsilon(1-\varepsilon)}\right) \sqrt{\frac{mm'}{l_{max}}}$$

queries to the oracle.

5.1.2 The Separation

We divide the separation into two parts.

Part One: *Something* **is in** $\mathsf{SQMA}_1^{\mathcal{A}}$

For every randomized-preimage-appropriate oracle $\mathcal{A}$, we show that a respective language $L_{\mathcal{A}}$ is in $\mathsf{SQMA}_1^{\mathcal{A}}$.

Theorem 5.8. For every randomized-preimage-appropriate language $L_{\mathcal{A}}$, which contains those unary strings 1^n such that

$$\mathscr{A}_n = \mathscr{P}_{\sigma_{\text{pre}}(S)}, \tag{5.16}$$

with $S \in \mathbf{S}^n_{\text{odd}}$, we have that $L_{\mathcal{A}}$ is in $\mathsf{SQMA}_1^{\mathcal{A}}$.

Proof. At first, we present a $\mathsf{SQMA}_1^{\mathcal{A}}$ protocol, and then we prove the completeness and soundness parts of the corresponding verification procedure.

The Protocol. Let $|\Sigma(S)\rangle$ denote the subset-state witness, put forth by the prover, when the oracle at hand is $\mathscr{P}_{\sigma_{\text{pre}}(S)}$, for some $S \in \mathbf{S}^n_{\text{odd}} \cup \mathbf{S}^n_{\text{bal}}$. The verifier performs the following tests, each with probability $1/2$:

Test 1. Apply $\mathscr{P}_{\sigma_{\text{pre}}(S)}$ to $|\Sigma(S)\rangle$, and then measure whether the resultant state is

$$|\Psi\rangle = \sum_{i=1}^{N} \frac{1}{\sqrt{N}} |i\rangle . \tag{5.17}$$

To do that, one can project in the subspace spanned by $|\Psi\rangle$, by employing the projector $\Pi_{|\Psi\rangle} = |\Psi\rangle\langle\Psi|$. Finally, note that $\| |\Psi\rangle \|_2 = 1$.

Test 2. Measure $|\Sigma(S)\rangle$ in the computational basis. Let i^* be the resulting standard basis state. If i^* is even, then output 0. Otherwise apply $\mathscr{P}_{\sigma_{\text{pre}}(S)}$ to $|i^*\rangle$ and measure in the standard basis. If the state is in the set $[N]$, then output 1, else output 0.

Completeness. Here we have that $1^n \in L_{\mathcal{A}}$, that is, one gets $\mathscr{A}_n = \mathscr{P}_{\sigma_{\text{pre}}(S)}$, with $S \in \mathbf{S}^n_{\text{odd}}$. Thus, we can use as a witness state the set S, that is,

$$|\Sigma(S)\rangle = |S\rangle . \tag{5.18}$$

If Test 1 is implemented, then the verifier will output 1 with probability 1. *Why?* The reason is that $\mathscr{P}_{\sigma_{\text{pre}}(S)}$ will transform $|S\rangle$ to $|\Psi\rangle$, as the Note 5.5 suggests. That is, the map $\mathscr{P}_{\sigma_{\text{pre}}(S)}$ applies a uniformly-at-random permutation $\sigma : S \to [N]$.

If Test 2 is implemented, then the verifier will output 1 with probability 1. *Why?* The reason is that the measured basis state i^* will be odd, and, moreover, after we apply to it the map $\mathscr{P}_{\sigma_{\text{pre}}(S)}$, we get something in $[N]$. See the Note 5.5.

Averaging over both tests, we get that the verifier accepts with probability equal to 1.

Soundness. Now suppose that $1^n \notin L_{\mathcal{A}}$. That is, $\mathscr{A}_n = \mathscr{P}_{\sigma_{\text{pre}}(S)}$, for some set $S \in \mathbf{S}^n_{\text{bal}}$.

We assume, for $\beta_i \in \mathbb{C}$, for every i, and $\sum_{i=1}^{N^2} |\beta_i|^2 = 1$, that the witness is

$$\begin{aligned} |\Sigma(S)\rangle &= \sum_{i=1}^{N^2} \beta_i |i\rangle \\ &= |\Sigma\rangle . \end{aligned} \tag{5.19}$$

Before we proceed, with the acceptance probability computation, we note that

$$\begin{aligned}\langle i|\,\mathcal{P}_\sigma^\dagger &= \langle \sigma(i)| \\ &= |\sigma(i)\rangle^\dagger \\ &= (\mathcal{P}_\sigma |i\rangle)^\dagger . \end{aligned} \tag{5.20}$$

Now, the probability that the verifier will output 1, after performing Test 1, is

$$\begin{aligned}
p_i &= \operatorname{Tr}\left(\Pi_{|\Psi\rangle}\rho\right) \\
&= \operatorname{Tr}\left(\Pi_{|\Psi\rangle}\mathscr{P}_{\sigma_{\mathrm{pre}}(S)}\left(|\Sigma\rangle\langle\Sigma|\right)\right) \\
&= \operatorname{Tr}\left(\Pi_{|\Psi\rangle}\frac{1}{|\sigma_{\mathrm{pre}}(S)|}\sum_{\sigma\in\sigma_{\mathrm{pre}}(S)}\mathcal{P}_\sigma|\Sigma\rangle\langle\Sigma|\,\mathcal{P}_\sigma^\dagger\right) \\
&= \operatorname{Tr}\left(\Pi_{|\Psi\rangle}\frac{1}{|\sigma_{\mathrm{pre}}(S)|}\sum_{\sigma\in\sigma_{\mathrm{pre}}(S)}\mathcal{P}_\sigma|\Sigma\rangle\langle\Sigma|\,\mathcal{P}_\sigma^\dagger\right) \\
&= \operatorname{Tr}\left(\Pi_{|\Psi\rangle}\frac{1}{|\sigma_{\mathrm{pre}}(S)|}\sum_{\sigma\in\sigma_{\mathrm{pre}}(S)}\mathcal{P}_\sigma\left(\sum_{i=1}^{N^2}\beta_i|i\rangle\right)\left(\sum_{j=1}^{N^2}\beta_j^*\langle j|\right)\mathcal{P}_\sigma^\dagger\right) \\
&= \operatorname{Tr}\left(\Pi_{|\Psi\rangle}\frac{1}{|\sigma_{\mathrm{pre}}(S)|}\sum_{\sigma\in\sigma_{\mathrm{pre}}(S)}\left(\sum_{i=1}^{N^2}\beta_i|\sigma(i)\rangle\right)\left(\sum_{j=1}^{N^2}\beta_j^*\langle\sigma(j)|\right)\right) \\
&= \operatorname{Tr}\left(\Pi_{|\Psi\rangle}\frac{1}{|\sigma_{\mathrm{pre}}(S)|}\sum_{\sigma\in\sigma_{\mathrm{pre}}(S)}\sum_{i,j=1}^{N^2}\beta_i\beta_j^*|\sigma(i)\rangle\langle\sigma(j)|\right) \\
&= \frac{1}{|\sigma_{\mathrm{pre}}(S)|}\operatorname{Tr}\left(\Pi_{|\Psi\rangle}\sum_{\sigma\in\sigma_{\mathrm{pre}}(S)}\sum_{i,j=1}^{N^2}\beta_i\beta_j^*|\sigma(i)\rangle\langle\sigma(j)|\right) \\
&= \frac{1}{|\sigma_{\mathrm{pre}}(S)|}\operatorname{Tr}\left(|\Psi\rangle\langle\Psi|\sum_{\sigma\in\sigma_{\mathrm{pre}}(S)}\sum_{i,j=1}^{N^2}\beta_i\beta_j^*|\sigma(i)\rangle\langle\sigma(j)|\right) \\
&= \frac{1}{|\sigma_{\mathrm{pre}}(S)|}\operatorname{Tr}\left(\left(\sum_{k=1}^{N}\frac{1}{\sqrt{N}}|k\rangle\right)\left(\sum_{m=1}^{N}\frac{1}{\sqrt{N}}\langle m|\right)\sum_{\sigma\in\sigma_{\mathrm{pre}}(S)}\sum_{i,j=1}^{N^2}\beta_i\beta_j^*|\sigma(i)\rangle\langle\sigma(j)|\right) \\
&= \frac{1}{N|\sigma_{\mathrm{pre}}(S)|}\operatorname{Tr}\left(\left(\sum_{k=1}^{N}|k\rangle\right)\left(\sum_{m=1}^{N}\langle m|\right)\sum_{\sigma\in\sigma_{\mathrm{pre}}(S)}\sum_{i,j=1}^{N^2}\beta_i\beta_j^*|\sigma(i)\rangle\langle\sigma(j)|\right) \\
&= \frac{1}{N|\sigma_{\mathrm{pre}}(S)|}\operatorname{Tr}\left(\sum_{k=1}^{N}|k\rangle\cdot\sum_{m=1}^{N}\langle m|\cdot\sum_{\sigma\in\sigma_{\mathrm{pre}}(S)}\sum_{i,j=1}^{N^2}\beta_i\beta_j^*|\sigma(i)\rangle\langle\sigma(j)|\right)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{N\left|\boldsymbol{\sigma}_{\text{pre}}(S)\right|}\text{Tr}\left(\sum_{k,m=1}^{N}|k\rangle\langle m|\cdot\sum_{\sigma\in\boldsymbol{\sigma}_{\text{pre}}(S)}\sum_{i,j=1}^{N^2}\beta_i\beta_j^*\left|\sigma(i)\right\rangle\left\langle\sigma(j)\right|\right)\\
&= \frac{1}{N\left|\boldsymbol{\sigma}_{\text{pre}}(S)\right|}\text{Tr}\left(\sum_{k,m=1}^{N}\sum_{\sigma\in\boldsymbol{\sigma}_{\text{pre}}(S)}\sum_{i,j=1}^{N^2}|k\rangle\langle m|\,\beta_i\beta_j^*\left|\sigma(i)\right\rangle\left\langle\sigma(j)\right|\right)\\
&= \frac{1}{N\left|\boldsymbol{\sigma}_{\text{pre}}(S)\right|}\text{Tr}\left(\sum_{k,m=1}^{N}\sum_{\sigma\in\boldsymbol{\sigma}_{\text{pre}}(S)}\sum_{i,j=1}^{N^2}\beta_i\beta_j^*\,|k\rangle\langle m|\left|\sigma(i)\right\rangle\left\langle\sigma(j)\right|\right)\\
&= \frac{1}{N\left|\boldsymbol{\sigma}_{\text{pre}}(S)\right|}\text{Tr}\left(\sum_{k,m=1}^{N}\sum_{\sigma\in\boldsymbol{\sigma}_{\text{pre}}(S)}\sum_{i,j=1}^{N^2}\beta_i\beta_j^*\,|k\rangle\left\langle m|\sigma(i)\right\rangle\left\langle\sigma(j)\right|\right),\quad(5.21)
\end{aligned}
$$

where, in order to get $\langle m|\sigma(i)\rangle = 1$, and not $\langle m|\sigma(i)\rangle = 0$, we see that $m = \sigma(i)$, or $i \in S$, since $m \in [N]$, and $\sigma : S \to [N]$. Next, for $i \in S$, we have that

$$
\begin{aligned}
p_i &= \frac{1}{N\left|\boldsymbol{\sigma}_{\text{pre}}(S)\right|}\text{Tr}\left(\sum_{k=1}^{N}\sum_{\sigma\in\boldsymbol{\sigma}_{\text{pre}}(S)}\sum_{i\in S}\beta_i\sum_{j=1}^{N^2}\beta_j^*\,|k\rangle\left\langle\sigma(j)\right|\right)\\
&= \frac{1}{N\left|\boldsymbol{\sigma}_{\text{pre}}(S)\right|}\sum_{k=1}^{N}\sum_{\sigma\in\boldsymbol{\sigma}_{\text{pre}}(S)}\sum_{i\in S}\beta_i\sum_{j=1}^{N^2}\beta_j^*\text{Tr}\left(|k\rangle\left\langle\sigma(j)\right|\right)\\
&= \frac{1}{N\left|\boldsymbol{\sigma}_{\text{pre}}(S)\right|}\sum_{k=1}^{N}\sum_{\sigma\in\boldsymbol{\sigma}_{\text{pre}}(S)}\sum_{i\in S}\beta_i\sum_{j=1}^{N^2}\beta_j^*\left\langle k|\sigma(j)\right\rangle,\quad(5.22)
\end{aligned}
$$

where, in order to get $\langle k|\sigma(j)\rangle = 1$, and not $\langle k|\sigma(j)\rangle = 0$, we see that $k = \sigma(j)$, or $j \in S$, since $k \in [N]$, and $\sigma : S \to [N]$. Finally, for $j \in S$, we have that

$$
\begin{aligned}
p_i &= \frac{1}{N\left|\boldsymbol{\sigma}_{\text{pre}}(S)\right|}\sum_{\sigma\in\boldsymbol{\sigma}_{\text{pre}}(S)}\sum_{i\in S}\beta_i\sum_{j\in S}\beta_j^*\\
&= \frac{1}{N\left|\boldsymbol{\sigma}_{\text{pre}}(S)\right|}\sum_{\sigma\in\boldsymbol{\sigma}_{\text{pre}}(S)}\sum_{i,j\in S}\beta_i\beta_j^*\\
&= \frac{1}{N\left|\boldsymbol{\sigma}_{\text{pre}}(S)\right|}\left|\boldsymbol{\sigma}_{\text{pre}}(S)\right|\sum_{i,j\in S}\beta_i\beta_j^*\\
&= \frac{1}{N}\sum_{i,j\in S}\beta_i\beta_j^*\\
&= \frac{1}{N}\sum_{i\in S}\beta_i\sum_{j\in S}\beta_j^*\\
&= \frac{1}{N}\sum_{i\in S}\beta_i\left(\sum_{j\in S}\beta_j\right)^*
\end{aligned}
$$

$$= \frac{1}{N}\left|\sum_{i\in S}\beta_i\right|^2 . \tag{5.23}$$

In the case that Test 2 is implemented, the probability that the verifier will output 1 is

$$p_{ii} = \sum_{i\in\mathbb{Z}_{\text{odd}}\cap S}|\beta_i|^2 , \tag{5.24}$$

because in order for the verifier to accept, it must, at first, measure an odd basis state i, and, also, this state i has to be in S in order for the secret permutation, of the CPTP map, to send it in the desired set $[N]$. Now, calculations! We have that

$$\begin{aligned}1 &= \sum_{i\in S}|\beta_i|^2 + \sum_{i\notin S}|\beta_i|^2\\ &= \frac{1}{\sqrt{2}}\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}|\beta_i|^2 + \frac{\sqrt{2}-1}{\sqrt{2}}\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}|\beta_i|^2\\ &\quad + \sum_{i\in\mathbb{Z}_{\text{odd}}\cap S}|\beta_i|^2 + \sum_{i\notin S}|\beta_i|^2\\ &= \frac{1}{\sqrt{2}}\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}|\beta_i|^2 + \sum_{i\in\mathbb{Z}_{\text{odd}}\cap S}|\beta_i|^2\\ &\quad + \frac{\sqrt{2}-1}{\sqrt{2}}\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}|\beta_i|^2 + \sum_{i\notin S}|\beta_i|^2 .\end{aligned} \tag{5.25}$$

By applying the Cauchy-Schwarz inequality, we get that

$$\left(\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}1^2\right)\left(\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}|\beta_i|^2\right) \geq \left|\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}\beta_i\right|^2 ,$$

or

$$\frac{N}{2}\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}|\beta_i|^2 \geq \left|\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}\beta_i\right|^2 ,$$

or

$$\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}|\beta_i|^2 \geq \frac{2}{N}\left|\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}\beta_i\right|^2 ,$$

or

$$\sqrt{\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}|\beta_i|^2} \geq \sqrt{\frac{2}{N}}\left|\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}\beta_i\right| . \tag{5.26}$$

Of course,

$$\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}|\beta_i|^2 \geq \sqrt{\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}|\beta_i|^2}$$
$$\geq \sqrt{\frac{2}{N}}\left|\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}\beta_i\right|, \tag{5.27}$$

and

$$\sum_{i\in\mathbb{Z}_{\text{odd}}\cap S}|\beta_i|^2 \geq \sqrt{\sum_{i\in\mathbb{Z}_{\text{odd}}\cap S}|\beta_i|^2}$$
$$\geq \sqrt{\frac{2}{N}}\left|\sum_{i\in\mathbb{Z}_{\text{odd}}\cap S}\beta_i\right|. \tag{5.28}$$

So,

$$\begin{aligned}
1 &= \sum_{i\in S}|\beta_i|^2 + \sum_{i\notin S}|\beta_i|^2 \\
&= \frac{1}{\sqrt{2}}\sum_{i\in\mathbb{Z}_{\text{odd}}\cap S}|\beta_i|^2 + \sum_{i\in\mathbb{Z}_{\text{even}}\cap S}|\beta_i|^2 \\
&\quad + \frac{\sqrt{2}-1}{\sqrt{2}}\sum_{i\in\mathbb{Z}_{\text{odd}}\cap S}|\beta_i|^2 + \sum_{i\notin S}|\beta_i|^2 \\
&\geq \sqrt{\frac{1}{N}}\left|\sum_{i\in\mathbb{Z}_{\text{odd}}\cap S}\beta_i\right| + \sqrt{\frac{2}{N}}\left|\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}\beta_i\right| \\
&\quad + \frac{\sqrt{2}-1}{\sqrt{2}}\sum_{i\in\mathbb{Z}_{\text{odd}}\cap S}|\beta_i|^2 + \sum_{i\notin S}|\beta_i|^2 \\
&\geq \sqrt{\frac{1}{N}}\left|\sum_{i\in S}\beta_i\right| + \frac{\sqrt{2}-1}{\sqrt{2}}\sum_{i\in\mathbb{Z}_{\text{odd}}\cap S}|\beta_i|^2 + \sum_{i\notin S}|\beta_i|^2 \\
&\geq \sqrt{\frac{1}{N}}\left|\sum_{i\in S}\beta_i\right| + \frac{\sqrt{2}-1}{\sqrt{2}}\sum_{i\in\mathbb{Z}_{\text{odd}}\cap S}|\beta_i|^2 \\
&= \sqrt{\frac{1}{N}}\sqrt{Np_i} + \frac{\sqrt{2}-1}{\sqrt{2}}p_{ii} \\
&= \sqrt{p_i} + \frac{\sqrt{2}-1}{\sqrt{2}}p_{ii},
\end{aligned} \tag{5.29}$$

or, equivalently,

$$1 \geq \sqrt{p_i} + \frac{\sqrt{2}-1}{\sqrt{2}}p_{ii},$$

or

$$\sqrt{p_i} \leq 1 - \frac{\sqrt{2}-1}{\sqrt{2}} p_{ii},$$

or

$$p_i \leq \left(1 - \frac{\sqrt{2}-1}{\sqrt{2}} p_{ii}\right)^2. \tag{5.30}$$

Thus, the average probability the verifier outputs 1 is

$$\frac{1}{2}\left(p_i + p_{ii}\right) \leq \frac{1}{2}\left(\left(1 - \frac{\sqrt{2}-1}{\sqrt{2}} p_{ii}\right)^2 + p_{ii}\right). \tag{5.31}$$

Taking the derivative of the right-hand side, we see that it is positive for every $p_{ii} \in [0,1]$, and an ascending function on p_{ii}, so, in order to maximize the right-hand side we take the value $p_{ii} = 1$. Thus, one is able to observe that

$$\frac{1}{2}\left(p_i + p_{ii}\right) \leq \frac{3}{4} = 0.75. \tag{5.32}$$

Now, we note that $0.75 \ll 1$, which means that there is a gap between the acceptance probabilities in the completeness and soundness parts of the verification procedure. Thus, by employing standard error-reduction techniques [80] we can put this problem in $\mathsf{SQMA}_1^{\mathcal{A}}$. □

Part Two: *Something* **is not in** $\mathsf{QCMA}^{\mathcal{A}}$

In this subsection, we show that there exists some randomized-preimage-appropriate oracle $\mathcal{A}$, such that $L_{\mathcal{A}}$ is not in $\mathsf{QCMA}^{\mathcal{A}}$. We do this by first reducing our problem to a simpler, yet in a sense equivalent, problem, which is suitable for applying the template, about obtaining oracle separations against QCMA, introduced by Fefferman and Kimmel.

Definition 5.9 (Preimage-appropriate Oracles)**.** Let $\mathcal{A}$ be a countably-infinite set of unitaries, that is,

$$\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \dots\}, \tag{5.33}$$

where each $\mathcal{A}_n$ implements a unitary operation on $2n$ qubits. We say that the oracle $\mathcal{A}$ is preimage-appropriate, if, for every natural n, one has that

$$\mathcal{A}_n = \mathcal{P}_\sigma, \tag{5.34}$$

for some permutation σ such that $S_{\text{pre}}(\sigma) \in \mathbf{S}^n_{\text{odd}} \cup \mathbf{S}^n_{\text{bal}}$.

Theorem 5.10. Given a randomized-preimage-appropriate oracle, namely $\mathcal{A}$, let $1^n \in L_{\mathcal{A}}$, if

$$\mathscr{A}_n = \mathscr{P}_{\sigma_{\text{pre}}(S)}, \tag{5.35}$$

for $S \in \mathbf{S}^n_{\text{odd}}$. Given a preimage-appropriate oracle $\tilde{\mathcal{A}}$, let $1^n \in L_{\mathcal{A}}$, if

$$\mathcal{A}_n = \mathcal{P}_\sigma, \tag{5.36}$$

for σ such that $S_{\text{pre}}(\sigma) \in \mathbf{S}^n_{\text{odd}}$. Then, if there exists a $\mathsf{QCMA}^{\mathcal{A}}$-machine, namely M, that decides $L_{\mathcal{A}}$, for every randomized-preimage-appropriate oracle $\mathcal{A}$, then there is a $\mathsf{QCMA}^{\tilde{\mathcal{A}}}_{\mathsf{EXP}}$-machine, namely $\tilde{M}$, that decides $L_{\tilde{\mathcal{A}}}$, for every preimage-appropriate oracle $\tilde{\mathcal{A}}$, such that $\tilde{M}$ uses at most a polynomial number of queries to $\tilde{\mathcal{A}}$, and, on input 1^n, $\tilde{M}$ takes as input a classical witness w that depends only on the set $S_{\text{pre}}(\sigma)$.

For the proof of Theorem 5.10, see the work of Fefferman and Kimmel [27]. We will now show that $L_{\mathcal{A}} \notin \mathsf{QCMA}^{\mathcal{A}}_{\mathsf{EXP}}$, by using the template introduced by Fefferman and Kimmel.

Lemma 5.11. There exists a preimage-appropriate oracle $\mathcal{A}$, such that there is no $\mathsf{QCMA}^{\mathcal{A}}_{\mathsf{EXP}}$-machine that decides $L_{\mathcal{A}}$, using a polynomial number of queries, where the classical witness, on input 1^n, for $n \in \mathbb{N}$, depends only on $S_{\text{pre}}(\sigma)$, for $\mathcal{A}_n = \mathcal{P}_\sigma$.

To prove Lemma 5.11, we use the aforementioned template which consists of four criteria, that are presented, and subsequently proved, below.

Lemma 5.12 (*Criterion 1*)**.** The oracle must be of the form

$$\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \dots\} \tag{5.37}$$

where each $\mathcal{A}_n$ implements a $p_1(n)$-qubit unitary, for $p_1(n)$ being a polynomial in $n \in \mathbb{N}$, and $\mathcal{U} \in \mathbf{U}^n = \mathbf{U}^n_X \cup \mathbf{U}^n_Y$, where $\mathbf{U}^n_X$ and $\mathbf{U}^n_Y$ are disjoint families of unitaries on $p_1(n)$ qubits. To each unitary $\mathcal{U}$ in $\mathbf{U}^n$ is associated a (not necessarily unique) subset

$$S_U(\mathcal{U}) \subseteq \left[2^{p_2(n)}\right], \tag{5.38}$$

for $p_2(n)$ a polynomial in $n \in \mathbb{N}$. We further require that

$$\mathbf{S}^n_X = \{S \mid S = S_U(\mathcal{U}) \text{ and } \mathcal{U} \in \mathbf{U}^n_X\} \tag{5.39}$$

is disjoint from

$$\mathbf{S}^n_Y = \{S \mid S = S_U(\mathcal{U}) \text{ and } \mathcal{U} \in \mathbf{U}^n_Y\}. \tag{5.40}$$

Finally, the language $L_{\mathcal{A}}$ must be such that $1^n \in L_{\mathcal{A}}$, if and only if $\mathcal{A}_n \in \mathbf{U}_X^n$.

Proof. In our case, *Criterion 1* is true for

$$p_1(n) = 2n, \tag{5.41}$$

$$p_2(n) = 2n, \tag{5.42}$$

$$\mathbf{S}_X^n = \mathbf{S}_{\text{odd}}^n, \tag{5.43}$$

$$\mathbf{S}_Y^n = \mathbf{S}_{\text{bal}}^n, \tag{5.44}$$

$$\mathbf{U}_X^n = \left\{\mathcal{P}_\sigma \mid S_{\text{pre}}(\sigma) \in \mathbf{S}_{\text{odd}}^n\right\}, \tag{5.45}$$

and

$$\mathbf{U}_Y^n = \left\{\mathcal{P}_\sigma \mid S_{\text{pre}}(\sigma) \in \mathbf{S}_{\text{bal}}^n\right\}. \tag{5.46}$$

□

Lemma 5.13 (*Criterion 2*)**.** The optimal witness to a $\mathsf{QCMA}^{\mathcal{A}}$ machine, that decides a language $L_{\mathcal{A}}$, as in *Criterion 1*, on input 1^n, must depend only on the set $S_U(\mathcal{U})$, where $\mathcal{U} = \mathcal{A}_n$.

Proof. Here, we only consider witnesses that depend on the set $S_{\text{pre}}(\sigma)$, which is the set $S_U(\mathcal{U})$, for $\mathcal{U} = \mathcal{A}_n = \mathcal{P}_\sigma$. □

Before we present, and prove, *Criterion 3*, we need the Definition 5.14.

Definition 5.14 (Distributed Classes of Sets)**.** We define a class of sets $\mathbf{S}$ to be $(\beta, \mathbf{S}_1)$*-distributed*, if satisfies the following three conditions:

1. there exists a set S_{fixed}, such that $S_{\text{fixed}} \subseteq S$ for all $S \in \mathbf{S}$,
2. there exists a subclass $\mathbf{S}' \subseteq \mathbf{S}_1$, such that $S_{\text{fixed}} \subseteq S$ for all $S \in \mathbf{S}'$, and
3. for every element i, such that $i \in \left(\bigcup_{S \in \mathbf{S}} S\right) \setminus S_{\text{fixed}}$, we have that the element i appears in at most a $N^{-\beta}$-fraction of the sets $S \in \mathbf{S}$.

Lemma 5.15 (*Criterion 3*)**.** Let $\alpha \in (0, 1/2)$ be some constant, and let $p(n)$ be a polynomial function in n. Then, there exists a positive integer $n^*(p(n), \alpha)$, such that for every $n > n^*(p(n), \alpha)$, and every subset $\mathbf{S} \subseteq \mathbf{S}_{\text{odd}}^n$, with

$$|\mathbf{S}| \geq \left|\mathbf{S}_{\text{odd}}^n\right| 2^{-p(n)}, \tag{5.47}$$

there is a subset $\mathbf{S}' \subseteq \mathbf{S}$ such that is $\left(\alpha, \mathbf{S}_{\text{bal}}^n\right)$-distributed.

Proof. We construct the desired $\mathbf{S}'$, by following the Fixing Procedure of Fefferman and Kimmel [27]. The Fixing Procedure is illustrated, below, in the Algorithm 4.

Algorithm 4 The Fixing Procedure.

```
1:  procedure FIXINGPROCEDURE($\mathbf{S}$)
2:      $\mathbf{S}' \leftarrow \mathbf{S}$
3:      $S_{\text{fixed}} \leftarrow \emptyset$
4:      while $\exists i \in \bigcup_{S \in \mathbf{S}'} \mathbf{S}' : i$ is odd do
5:          $v(i) \leftarrow$ the number of sets $S \in \mathbf{S}'$, such that $i \in S$
6:          if $|\mathbf{S}'| > v(i) \geq |\mathbf{S}'| N^{-\alpha}$ then
7:              $\mathbf{S}' \leftarrow \{S \mid S \in \mathbf{S}' \text{ and } i \in S\}$
8:              $S_{\text{fixed}} \leftarrow S_{\text{fixed}} \cup \{i\}$
9:          end if
10:     end while
11:     return $\mathbf{S}'$
12: end procedure
```

By construction, $\mathbf{S}'$ satisfies the conditions 1 and 3, of the Definition 5.14. That is, the condition 1 is met because we explicitly create such a common subset S_{fixed}, and the condition 2 is met because we fix all the odd elements that appear in many sets, that is, in more than a specific portion of sets. We now have to prove that $\mathbf{S}'$ will also satisfy the condition 2.

We naturally expect that S_{fixed} has at most $N/2$ odd elements, since it is a subset of every set S of some subclass of the class $\mathbf{S}^n_{\text{bal}}$, because $\mathbf{S}'$ is $(\alpha, \mathbf{S}^n_{\text{bal}})$-distributed, and, also, S_{fixed} is expected to have zero even elements, since it is a subset of every set S in the class $\mathbf{S}^n_{\text{odd}}$.

Okay, the truthfulness of the fact that S_{fixed} has zero even elements is clearly visible from the Fixing Procedure, since we only fix odd elements.

Now, suppose that, at some point, we have $N/2$ odd elements in the common set S_{fixed}, and, simultaneously, an odd element i^* appears in a greater than a $N^{-\alpha}$-fraction of the sets in the class $\mathbf{S}'$. What we want is to reach at a contradiction. In this way we will show that this element i^* is *not* going to be chosen, as a member of the set S_{fixed}, by our Fixing Procedure. This would imply the truthfulness of the desired property $|S_{\text{fixed}}| \leq N/2$.

We now define the set

$$\mathbf{S}'' = \{S \mid S \in \mathbf{S}' \text{ and } i^* \in S\}. \tag{5.48}$$

Since $(S_{\text{fixed}} \cup \{i^*\}) \subseteq S$ for all $S \in \mathbf{S}''$, there are $N/2 - 1$ odd elements that can be freely chosen from the remaining $N^2/2 - N/2 - 1$ odd elements, for each set in $\mathbf{S}''$. Thus, we have that

$$\left|\mathbf{S}''\right| \leq \binom{N^2/2 - N/2 - 1}{N/2 - 1}. \tag{5.49}$$

By our assumption, regarding the number of sets that contain i^*, we have that

$$\left|\mathbf{S}''\right| \geq \left|\mathbf{S}'\right| N^{-\alpha}. \tag{5.50}$$

So,

$$\begin{aligned}
\left|\mathbf{S}'\right| &\leq \binom{N^2/2 - N/2 - 1}{N/2 - 1} N^{\alpha} \\
&\leq \binom{N^2/2}{N/2} N^{\alpha} \\
&\leq \left(\frac{N^2/2}{N/2} e\right)^{N/2} N^{\alpha} \\
&\leq (Ne)^{N/2} N^{\alpha} \\
&= 2^{N/2 \log(Ne) + \alpha \log N} \\
&= 2^{\mathcal{O}(N) + (N/2) \log N}. \qquad (5.51)
\end{aligned}$$

This is a bound on $|\mathbf{S}'|$. It turns out that one can devise yet another bound on this quantity. To do this, one takes into consideration the fact that

$$\left|\mathbf{S}'\right| \geq |\mathbf{S}^n_{\text{odd}}| \, 2^{-p(n)}, \tag{5.52}$$

straight from our lemma-hypothesis, and the fact that, in each cycle of the Fixing Procedure, the size of the set $|\mathbf{S}'|$ is reduced at most by a factor equal to $N^{-\alpha}$. So, if we take into consideration that the Fixing Procedure runs for at most $N/2$ cycles, we have that

$$\begin{aligned}
\left|\mathbf{S}'\right| &\geq |\mathbf{S}^n_{\text{odd}}| \, 2^{-p(n)} N^{-\alpha N/2} \\
&= \binom{N^2/2}{N} 2^{-p(n)} N^{-\alpha N/2} \\
&\geq \left(\frac{N^2/2}{N}\right)^{N} 2^{-p(n)} N^{-\alpha N/2} \\
&= (N/2)^{N} 2^{-p(n)} N^{-\alpha N/2} \\
&= 2^{N \log(N/2) - p(\log N) - \alpha N/2 \log N} \\
&= 2^{-\mathcal{O}(N) + N \log(N) - \alpha N/2 \log N} \\
&= 2^{-\mathcal{O}(N) + (N/2)(2-\alpha) \log N}. \qquad (5.53)
\end{aligned}$$

We observe that the bound of (5.53) dominates that of (5.51), for every value of the constant $\alpha \in (0, 1/2)$, and for large enough values of N, that is, for values of N satisfying $N > 2^{n^*}$, where n^* depends only on $p(n)$ and α. This is, of course, a contradiction, and so all odd elements out of S_{fixed} will be contained in at most a $N^{-\alpha}$-fraction of $S \in \mathbf{S}'$. Hence, at the next cycle of the Fixing Procedure, this odd element i^* will not be added to S_{fixed}, and the number of odd elements in S_{fixed} will remain bounded by $N/2$. The same logic can be of course reapplied to future cycles of the Fixing Procedure, guaranteeing this desired property!

The condition 2 is now met, since the fact $|S_{\text{fixed}}| \leq N/2$ implies that there truly exists some subclass of $\mathbf{S}_{\text{bal}}^n$, such that S_{fixed} is a subset of each of its sets. □

Lemma 5.16 (*Criterion 4*)**.** Suppose that $\mathbf{S}_X \subseteq \mathbf{S}_{\text{odd}}^n$ is $\left(\delta, \mathbf{S}_{\text{bal}}^n\right)$-distributed, for some positive δ. Then, for every quantum algorithm G, there exists a permutation σ_x, such that $S_{\text{pre}}(\sigma_x) \in \mathbf{S}_{\text{odd}}^n$, and a permutation σ_y, such that $S_{\text{pre}}(\sigma_y) \in \mathbf{S}_{\text{bal}}^n$, such that given oracle access to either $\mathcal{P}_{\sigma_x}$ or $\mathcal{P}_{\sigma_y}$, the quantum algorithm G cannot distinguish them with probability at least ε, without using at least

$$\left(1 - 2\sqrt{\varepsilon(1-\varepsilon)}\right) N^{\delta/2}$$

queries.

Proof. Since $\mathbf{S}_X \subseteq \mathbf{S}_{\text{odd}}^n$ is $\left(\delta, \mathbf{S}_{\text{bal}}^n\right)$-distributed, there exists a set S_{fixed} such that $S_{\text{fixed}} \subseteq S$, for all S in $\mathbf{S}_X$. Thus, S_{fixed} contains at most $N/2$ odd elements, since it is $\left(\delta, \mathbf{S}_{\text{bal}}^n\right)$-distributed and, therefore, there are some sets S in $\mathbf{S}_{\text{bal}}^n$ that contain it as a subset, and, also, contains no even elements, since it is contained in every $S \in \mathbf{S}_X \subseteq \mathbf{S}_{\text{odd}}^n$.

We now define

$$\mathbf{S}_Y = \left\{S \mid S \in \mathbf{S}_{\text{bal}}^n \text{ and } S_{\text{fixed}} \subseteq S\right\}, \tag{5.54}$$

$$\begin{aligned}\boldsymbol{\sigma}_X &= \left\{\sigma \mid S_{\text{pre}}(\sigma) \in \mathbf{S}_X\right\} \\ &= \bigcup_{S_x \in \mathbf{S}_X} \boldsymbol{\sigma}_{\text{pre}}(S_x),\end{aligned} \tag{5.55}$$

and

$$\boldsymbol{\sigma}_Y = \left\{\sigma \mid S_{\text{pre}}(\sigma) \in \mathbf{S}_Y\right\}$$

$$= \bigcup_{S_y \in \mathbf{S}_Y} \boldsymbol{\sigma}_{\text{pre}}\left(S_y\right). \tag{5.56}$$

We are now going to define a relation $\mathbf{R} \subseteq \boldsymbol{\sigma}_X \times \boldsymbol{\sigma}_Y$, that we will later on employ to apply our quantum adversary bound argument.

For every $\left(S_x, S_y\right) \in \mathbf{S}_X \times \mathbf{S}_Y$, we will create a one-to-one matching in the relation $\mathbf{R}$ between the elements of $\boldsymbol{\sigma}_{\text{pre}}\left(S_x\right)$ and $\boldsymbol{\sigma}_{\text{pre}}\left(S_x\right)$. We first choose any permutation $\sigma_x^* \in \boldsymbol{\sigma}_{\text{pre}}\left(S_x\right)$. Then, we seek for some desired permutation $\sigma_y^* \in \boldsymbol{\sigma}_{\text{pre}}\left(S_y\right)$ such that

1. $\forall j \in \left(S_x \cap S_y\right) : \sigma_x^*\left(j\right) = \sigma_y^*\left(j\right)$,
2. $\forall j \in \left[N^2\right] \setminus \left(S_x \cup S_y\right) : \sigma_x^*\left(j\right) = \sigma_y^*\left(j\right)$, and
3. $\forall j \in S_x \setminus \left(S_x \cap S_y\right), \exists i \in S_y \setminus \left(S_x \cap S_y\right) : \sigma_x^*\left(j\right) = \sigma_y^*\left(i\right)$, and
$$\sigma_x^*\left(i\right) = \sigma_y^*\left(j\right).$$

Since every permutation, with preimage S_y, is in $\boldsymbol{\sigma}_{\text{pre}}\left(S_y\right)$, there will always be a permutation σ_y^* such that satisfies the above three criteria. We now let

$$\left(\sigma_x^*, \sigma_y^*\right) \in \mathbf{R}. \tag{5.57}$$

For $i \in \left[N!\left(N^2 - N\right)!\right]$, let $\boldsymbol{\tau}^n = \{\tau_i\}_i \subseteq \boldsymbol{\sigma}^n$ be the set of all permutations, from $\left[N^2\right]$ to $\left[N^2\right]$, such that they do not mix the first N elements with the rest of the $N^2 - N$ elements.

Note 5.17. Note that we denote by $\left(a \circ b\right)\square$ the permutation obtained by first applying the permutation b and then the permutation a, on the "permutable" object $\square$.

Note 5.18. Note, also, that, for every pair of sets $\left(S_x, S_y\right) \in \mathbf{S}_X \times \mathbf{S}_Y$, one has that

$$\boldsymbol{\sigma}_{\text{pre}}\left(S_x\right) = \left\{\tau \circ \sigma_x^* \mid \tau \in \boldsymbol{\tau}^n\right\}, \tag{5.58}$$

and

$$\boldsymbol{\sigma}_{\text{pre}}\left(S_y\right) = \left\{\tau \circ \sigma_y^* \mid \tau \in \boldsymbol{\tau}^n\right\}. \tag{5.59}$$

The reason behind these equalities is that the permutations $\{\tau_i \circ \sigma\}_i$ do not alter the preimage of the permutation σ.

Given a permutation $\tau \in \boldsymbol{\tau}^n$, we have that

1. $\forall j \in \left(S_x \cap S_y\right) : \tau \circ \sigma_x^* (j) = \tau \circ \sigma_y^* (j)$,
2. $\forall j \in \left[N^2\right] \setminus \left(S_x \cup S_y\right) : \tau \circ \sigma_x^* (j) = \tau \circ \sigma_y^* (j)$, and
3. $\forall j \in S_x \setminus \left(S_x \cap S_y\right), \exists i \in S_y \setminus \left(S_x \cap S_y\right) : \tau \circ \sigma_x^* (j) = \tau \circ \sigma_y^* (i)$, and
$$\tau \circ \sigma_x^* (i) = \tau \circ \sigma_y^* (j) .$$

For every $\tau \in \boldsymbol{\tau}^n$, we let

$$\left(\tau \circ \sigma_x^*, \tau \circ \sigma_y^*\right) \in \mathbf{R}. \tag{5.60}$$

In this way, we are able to create a one-to-one matching between $\boldsymbol{\sigma}_{\text{pre}} (S_x)$ and $\boldsymbol{\sigma}_{\text{pre}} \left(S_y\right)$. We, then, repeat this process for every pair of sets $\left(S_x, S_y\right) \in \mathbf{S}_X \times \mathbf{S}_Y$, and this is the relation $\mathbf{R}$ we want!

Now we analyze this relation $\mathbf{R}$ we created.

Notice that each $\sigma_x \in \boldsymbol{\sigma}_X$ is paired to exactly one element σ_y of $\boldsymbol{\sigma}_{\text{pre}} \left(S_y\right)$, for each set $S_y \in \mathbf{S}_Y$. Thus, the number of the permutations σ_y matched to σ_x is equal to $m = |\mathbf{S}_Y|$. In a similar fashion, one has that the number of the permutations $\sigma_x \in \boldsymbol{\sigma}_{\text{pre}} (S_x)$ matched to any $\sigma_y \in \boldsymbol{\sigma}_Y$ is equal to $m' = |\mathbf{S}_X|$, since there is a unique σ_x matched to σ_y: one for every $S_x \in \mathbf{S}_X$.

We now consider a pair $\left(\sigma_x, \sigma_y\right) \in \mathbf{R}$. We, also, consider an element j such that

$$\sigma_x (j) \neq \sigma_y (j) , \tag{5.61}$$

and, then, we discriminate between two cases, regarding whether j is in S_x or S_y.

Case where $j \in S_x$. We upper bound $l_{x,j}$, that is, the number of the permutations $\sigma_{y'}$ such that $\left(\sigma_x, \sigma_{y'}\right) \in \mathbf{R}$. Since σ_x is matched to exactly one $\sigma_{y'}$, for each $S_{y'} \in \mathbf{S}_Y$, we have that

$$l_{x,j} \leq |\mathbf{S}_Y| . \tag{5.62}$$

We now upper bound $l_{y,j}$, that is, the number of permutations $\sigma_{x'}$ that are such that $\left(\sigma_{x'}, \sigma_y\right) \in \mathbf{R}$. By the construction of $\mathbf{R}$, and (5.61), we get that $j \notin S_y \supseteq S_{\text{fixed}}$. Also, by the construction of $\mathbf{R}$, we get that if $j \notin S_y$, and $\sigma_{x'} (j) \neq \sigma_y (j)$, then one has that $j \in S_{x'}$. Since σ_y is paired to exactly one $\sigma_{x'}$, for every set $S_{x'} \in \mathbf{S}_X$, the quantity $l_{y,j}$ is bounded above by the number of sets $S_{x'} \in \mathbf{S}_X$ which contain the number j. Since the set $\mathbf{S}_X$ is $\left(\delta, \mathbf{S}^n_{\text{bal}}\right)$-distributed, we get that at most a $N^{-\delta}$-fraction of the sets in $\mathbf{S}_X$ can contain j, since $j \notin S_{\text{fixed}}$. So, we get that

$$l_{y,j} \leq |\mathbf{S}_X|\, N^{-\delta}. \tag{5.63}$$

In total, we have that

$$l_{x,j} l_{y,j} \leq |\mathbf{S}_X|\, |\mathbf{S}_Y|\, N^{-\delta}. \tag{5.64}$$

Let us now look at the other case.

Case where $j \in S_y$. We upper bound $l_{y,j}$, that is, the number of $\sigma_{x'}$ such that $\left(\sigma_{x'}, \sigma_y\right) \in \mathbf{R}$, and $\sigma_{x'}(j) \neq \sigma_y(j)$. Since σ_y is paired to exactly one element $\sigma_{x'}$, for each $S_{x'} \in \mathbf{S}_X$, we have that

$$l_{y,j} \leq |\mathbf{S}_X|\,. \tag{5.65}$$

We now upper bound $l_{x,j}$, that is, the number of permutations $\sigma_{y'}$ that are such that $\left(\sigma_x, \sigma_{y'}\right) \in \mathbf{R}$, and $\sigma_x(j) \neq \sigma_{y'}(j)$. By the construction of $\mathbf{R}$, and (5.61), we have that $j \notin S_x \supseteq S_{\text{fixed}}$. Also, by the contruction of $\mathbf{R}$, if $j \notin S_x$, and $\sigma_x(j) \neq \sigma_{y'}(j)$, then $j \in S_{y'}$. Thus, since σ_x is paired to exactly one element $\sigma_{y'}$, for every $S_{y'}$, we have that $l_{x,j}$ is upper bounded by the number of the sets $S_{y'} \in \mathbf{S}_Y$ that contain the number $j \notin S_{\text{fixed}}$. Thus, if S_{fixed} has k_{odd} elements, and j is odd, we get that

$$\begin{aligned} l_{x,j} &= \binom{N^2/2 - k_{\text{odd}} - 1}{N/2 - k_{\text{odd}} - 1} \binom{N^2/2}{N/2} \\ &\leq \frac{N/2}{N^2/2 - N/2} \binom{N^2/2 - k_{\text{odd}}}{N/2 - k_{\text{odd}}} \binom{N^2/2}{N/2} \\ &= \frac{N/2}{N^2/2 - N/2} |\mathbf{S}_Y|\,. \end{aligned} \tag{5.66}$$

In the case that $j \in S_{y'}$ is even, we get that

$$\begin{aligned} l_{x,j} &= \binom{N^2/2 - k_{\text{odd}}}{N/2 - k_{\text{odd}}} \binom{N^2/2 - 1}{N/2 - 1} \\ &\leq \frac{N/2}{N^2/2 - N/2} |\mathbf{S}_Y|\,. \end{aligned} \tag{5.67}$$

In the Equations (5.66) and (5.67), we used the fact that

$$|\mathbf{S}_Y| = \binom{N^2/2 - k_{\text{odd}}}{N/2 - k_{\text{odd}}} \binom{N^2/2}{N/2}, \tag{5.68}$$

which holds since each $S \in \mathbf{S}_Y$ is a superset of S_{fixed}, which contains only odd elements, and $|S_{\text{fixed}}| = k_{\text{odd}}$. Also, the one half of the elements of S are odd, and the other half of its elements are even. Therefore

$$\begin{aligned} l_{x,j} l_{y,j} &\leq \frac{N/2}{N^2/2 - N/2} |\mathbf{S}_X| \, |\mathbf{S}_Y| \\ &= |\mathbf{S}_X| \, |\mathbf{S}_Y| \, \mathcal{O}\left(N^{-1}\right). \end{aligned} \tag{5.69}$$

Resolution. Clearly, as one observes, the bound of (5.64) dominates the bound of (5.69), since $\delta < 1$, so we can write that

$$\begin{aligned} \sqrt{\frac{mm'}{l_{x,j} l_{y,j}}} &\geq \sqrt{\frac{|\mathbf{S}_X| \, |\mathbf{S}_Y|}{|\mathbf{S}_X| \, |\mathbf{S}_Y| \, N^{-\delta}}} \\ &= N^{\delta/2}. \end{aligned} \tag{5.70}$$

Thus, by employing Lemma 5.7, we get that for any quantum algorithm which makes T queries to an oracle $\mathcal{P}_{\sigma_x}$, where σ_x is promised to be either in $\boldsymbol{\sigma}_X$ or $\boldsymbol{\sigma}_Y$, then there exist at least one element of $\boldsymbol{\sigma}_X$ and at least one element of $\boldsymbol{\sigma}_Y$, such that, in order for the probability, that our quantum algorithm effectively distinguishes between the corresponding oracles, to be at least $1 - \varepsilon$, for $\varepsilon \geq 0$, one has that

$$T \geq \left(1 - 2\sqrt{\varepsilon(1-\varepsilon)}\right) \sqrt{\frac{mm'}{l_{max}}} \leq \left(1 - 2\sqrt{\varepsilon(1-\varepsilon)}\right) \sqrt{\frac{mm'}{l_{x,j} l_{y,j}}} \tag{5.71}$$

since $l_{max} \geq l_{x,j} l_{y,j}$, or

$$T \in \Omega\left(N^{\delta/2}\right), \tag{5.72}$$

in the worst case. This is equivalent to saying that there exists one element of $\boldsymbol{\sigma}_X$ and one element of $\boldsymbol{\sigma}_Y$ such that in order for G to distinguish them, with constant bias, G requires about $T \in \Omega\left(N^{\delta/2}\right)$ queries. □

We now present the theorem that we need to finish the separation. The proof of this theorem, Theorem 5.19, can be found in the work by Fefferman and Kimmel [27].

Theorem 5.19. Let $\mathcal{A}$ be a quantum oracle, and let $L_{\mathcal{A}}$ be a language that satisfies the Criteria 1, 2, 3, and 4. Then, the language $L_{\mathcal{A}}$ requires exponentially many queries, in order to be decided, for any $\mathsf{QCMA}^{\mathcal{A}}$-machine. That is, $L_{\mathcal{A}}$ is hard for any verification—Merlin-Quantum-Arthur-style—setting, where the classical proof-Merlin is of polynomial size. We get $L_{\mathcal{A}} \notin \mathsf{QCMA}^{\mathcal{A}}$. Equivalently, $L_{\mathcal{A}}$ is hard for any efficient quantum algorithm, that verifies short classical proofs, when it comes to query complexity.

Thus, since Theorem 5.19 applies, here, the language $L_{\mathcal{A}}$, for $\mathcal{A}$ being a preimage-appropriate oracle, cannot be decided in the class $\mathsf{QCMA}^{\mathcal{A}}$, by any $\mathsf{QCMA}^{\mathcal{A}}$-machine, *by using only polynomially-many queries*. Hence, the language $L_{\mathcal{A}}$ cannot be decided in $\mathsf{QCMA}^{\mathcal{A}}_{\mathsf{EXP}}$, by any $\mathsf{QCMA}^{\mathcal{A}}_{\mathsf{EXP}}$-machine, *by using only polynomially-many queries*, again. We get that every $\mathsf{QCMA}^{\mathcal{A}}_{\mathsf{EXP}}$-machine is useless, given it uses at most polynomially-many queries to the oracle. Finally, from Theorem 5.10, there is not any $\mathsf{QCMA}^{\tilde{\mathcal{A}}}$-machine, for $\tilde{\mathcal{A}}$ being a randomized-preimage-appropriate oracle, that decides $L_{\tilde{\mathcal{A}}}$. We are done!

Note 5.20. The reason that Theorem 5.19 works well, here, is that the classes $\mathsf{QCMA}^{\mathcal{A}}$ and $\mathsf{QCMA}^{\mathcal{A}}_{\mathsf{EXP}}$ are very similar, from a query complexity point of view. Their differences are in their time bounds, not in their ability to extract information from oracle queries.

Chapter 6

Discussion and Conclusion

It is our hope that quantum computers will come into existence during our lifetime and that they will be harnessed to computational tasks beyond the reach of even the fastest possible classical computers.

— André Berthiaume, and Gilles Brassard,
Oracle Quantum Computing (1994)

Okay, so we proved a quantum-oracle separation. What does that imply? What are the consequences? What does that mean for our world, and computational complexity? Is this result any useful at all?

6.1 Relativized Worlds and Our Reality

Undoubtedly, relativization brings to the table some weird consequences. By employing, appropriate each time, oracles we are sometimes able to prove things unattainable before, yet in cases disturbing, such as the result where there exists some oracle $\mathcal{A}$ such that

$$\mathsf{IP}^{\mathcal{A}} \neq \mathsf{PSPACE}^{\mathcal{A}}, \tag{6.1}$$

for IP denoting *interactive proofs*, although

$$\mathsf{IP} = \mathsf{PSPACE}. \tag{6.2}$$

Similarly, the result

$$\mathsf{QMA}^{\mathcal{A}} \not\subseteq \mathsf{QMA_1}^{\mathcal{A}}, \tag{6.3}$$

for some oracle $\mathcal{A}$, implies that

$$\mathsf{BQP}^{\mathcal{A}} \not\subseteq \mathsf{ZQEXP}^{\mathcal{A}}, \tag{6.4}$$

for ZQEXP denoting *zero-error quantum exponential-time*, since

$$\mathsf{BQP} \subseteq \mathsf{QMA}, \tag{6.5}$$

and

$$\begin{aligned} \mathsf{QMA}_1 &\subseteq \mathsf{EXP} \\ &\subseteq \mathsf{ZQEXP}. \end{aligned} \tag{6.6}$$

However,

$$\mathsf{BQP} \subseteq \mathsf{ZQEXP}, \tag{6.7}$$

and we get yet another alienating result.

From a methodological point of view, oracle results help us understand better the power of complexity classes, since they shed light on the classes in the realms of worlds where intriguing possibilities hold, like the existence of polynomial-time deterministic algorithms for problems like `VERTEX COVER`, or so.

6.2 About Our Result: What Have We Learned?

In this thesis, we studied some of the effects that relativized worlds exert on the body of quantum complexity theory. In particular, we studied several separations involving classical and quantum computational complexity classes, in various relativized settings, induced by classical or quantum oracles. At the end, we were able to prove a simple, yet novel, separation in an appropriate relativized world.

To be more precise, we showed that there exists some quantum oracle $\mathcal{A}$, such that

$$\mathsf{SQMA}_1^{\mathcal{A}} \not\subseteq \mathsf{QCMA}^{\mathcal{A}}. \tag{6.8}$$

In our proof, we thoroughly used the work by Fefferman and Kimmel [27], in which they proved that

$$\mathsf{SQMA}^{\mathcal{A}} \not\subseteq \mathsf{QCMA}^{\mathcal{A}}, \tag{6.9}$$

by introducing a smart template for obtaining oracle separations against the class QCMA. Note that

$$\exists \mathcal{A} : \mathsf{SQMA}_1^{\mathcal{A}} \not\subseteq \mathsf{QCMA}^{\mathcal{A}} \Rightarrow \exists \mathcal{A} : \mathsf{SQMA}^{\mathcal{A}} \not\subseteq \mathsf{QCMA}^{\mathcal{A}}, \tag{6.10}$$

which means that our result is somewhat stronger.

6.3 Future Work and Open Problems

This thesis, we also attempted to show that there exists some oracle $\mathcal{U}$, such that

$$\mathsf{QMA}_1^{\mathcal{U}} \not\subseteq \mathsf{SQMA}_1^{\mathcal{U}}. \tag{6.11}$$

However, we were not able to prove it, *yet*.

6.3.1 First Idea

Our main intuition, about the truthfulness of the Equation (6.11), is the fact that, for a quantum system on n qubits, there exist about

$$2^{2^n}$$

subset states, since we can use, or not, any of the 2^n possible basis states, whereas there exist about

$$|\mathbb{C}|^{2^n} \gg 2^{2^n}$$

pure states, in total. It is obvious that there exist far more pure states than subset-flavored ones. Thus, there *must be* some quantum cirquit-verifier which is not perfectly convinced by *any* subset state, that is put forth, as a purported proof, by any potential prover. *But this is not a proof!*

To formalize, a bit, our thoughts, we write the following about the closely related problem $\mathsf{QMA}_1^{\mathcal{U}[1]} \not\subseteq \mathsf{SQMA}_1^{\mathcal{U}[1]}$, that is, the problem where we allowed to make only one query to the oracle $\mathcal{U}$. Now, let, for some state $|\psi\rangle$, on n qubits, $U_{|\psi\rangle}$ be a quantum oracle [13] which somewhat conceals the state $|\psi\rangle$, as follows:

$$U_{|\psi\rangle} |\phi\rangle = \begin{cases} -|\psi\rangle & \text{if } |\phi\rangle = |\psi\rangle, \text{ and} \\ |\phi\rangle & \text{if } \langle\phi|\psi\rangle = 0, \end{cases} \tag{6.12}$$

We now create a family of quantum oracles $\mathcal{U}$, which is just a disjoint union of "good" and "bad" oracles, namely

$$\begin{aligned} \mathcal{U} &= \mathcal{U}_{\text{good}} \uplus \mathcal{U}_{\text{bad}} \\ &= \{U_r\}_r \uplus \{U_q\}_q \\ &= \{U_i\}_{i\in\mathbb{N}}, \end{aligned} \tag{6.13}$$

with, if we let I denote the identity transformation,

$$U_i = \begin{cases} U_{|\psi\rangle} & \text{for some } |\psi\rangle, \text{if the oracle } U_i \text{ is good, and} \\ I & \text{if the oracle } U_i \text{ is bad.} \end{cases} \tag{6.14}$$

The problem is this: having an oracle U_k, and some purported proof $|\psi\rangle$, about $U_k \in \mathcal{U}_{\text{good}}$, as an input, we want to decide whether U_k is a good, or a bad, oracle, by making use of both the oracle U_k, and the proof $|\psi\rangle$. We know that this problem is in $\mathsf{QMA}_1^{\mathcal{U}[1]}$ [13].

To show that this problem is not in $\mathsf{SQMA}_1^{\mathcal{U}[1]}$, we are now looking for two states, namely an acceptance and a rejection states, denoted by $|\phi_A\rangle$ and $|\phi_R\rangle$, respectively, such that, if we let Π_{acc} denote the projection to the acceptance subspace, that is, the space that contains vectors which make our veifier accept, we have

$$\begin{aligned}
&\Pr\left[\text{The verifier replies with } \texttt{Yes}\text{, in the } \texttt{Yes}\text{-case.}\right] \\
&\qquad = \|\Pi_{\text{acc}} |\phi_A\rangle\|_2^2 \\
&\qquad = 1,
\end{aligned} \tag{6.15}$$

and

$$\begin{aligned}
&\Pr\left[\text{The verifier replies with } \texttt{Yes}\text{, in the } \texttt{No}\text{-case.}\right] \\
&\qquad = \|\Pi_{\text{acc}} |\phi_R\rangle\|_2^2 \\
&\qquad \leq \varepsilon,
\end{aligned} \tag{6.16}$$

for some $\varepsilon > 0$. Note that the Equation (6.16) holds, by invoking some error reduction technique [80, 103]. About these states one can write, while letting the projector Π_0 denote the projector that projects to the valid-inputs subspace, letting V be some unitary transformation, and, finally, for some unitary $U_{\text{bad}} = I \in \mathcal{U}_{\text{bad}}$, that

$$\begin{aligned}
|\phi_R\rangle &= U_{\text{bad}} V \Pi_0 |S\rangle \\
&= I V \Pi_0 |S\rangle \\
&= V \Pi_0 |S\rangle \\
&= |\theta'\rangle
\end{aligned} \tag{6.17}$$

or, if we decompose the ket $|\theta'\rangle$ in the ket $|\theta\rangle$, and the last register $(\gamma_0 |0\rangle + \gamma_1 |1\rangle)$,

$$|\phi_R\rangle = |\theta\rangle \otimes (\gamma_0 |0\rangle + \gamma_1 |1\rangle) \tag{6.18}$$

or, if we decompose $|\theta\rangle$ in the orthonormal $\{|\psi\rangle, |\psi^\perp\rangle\}$ basis, in which the ket $|\psi^\perp\rangle$ denotes a vector orthogonal to $|\psi\rangle$,

$$|\phi_R\rangle = \left(\beta_0 |\psi\rangle + \beta_1 \left|\psi^\perp\right\rangle\right) \otimes (\gamma_0 |0\rangle + \gamma_1 |1\rangle) \tag{6.19}$$

or, if we let $\alpha_{ij} = \beta_i \cdot \gamma_j$, for every pair (i, j), we have that

$$\begin{aligned}
|\phi_R\rangle &= \alpha_{00} |\psi\rangle \otimes |0\rangle + \alpha_{01} |\psi\rangle \otimes |1\rangle + \alpha_{10} \left|\psi^\perp\right\rangle \otimes |0\rangle + \alpha_{11} \left|\psi^\perp\right\rangle \otimes |1\rangle \\
&= \alpha_{00} |\psi\rangle |0\rangle + \alpha_{01} |\psi\rangle |1\rangle + \alpha_{10} \left|\psi^\perp\right\rangle |0\rangle + \alpha_{11} \left|\psi^\perp\right\rangle |1\rangle \\
&= \alpha_{00} |\psi, 0\rangle + \alpha_{01} |\psi, 1\rangle + \alpha_{10} \left|\psi^\perp, 0\right\rangle + \alpha_{11} \left|\psi^\perp, 1\right\rangle, \qquad (6.20)
\end{aligned}$$

with

$$\begin{aligned}
\||\psi_R\rangle\|_2^2 &= \sum_{i \in \{0,1\}^2} |\alpha_i|^2 \\
&= 1, \qquad (6.21)
\end{aligned}$$

and, for $U_{\text{good}} = U_{|\psi\rangle} \in \mathcal{U}_{\text{good}}$, for some $|\psi\rangle$,

$$\begin{aligned}
|\phi_A\rangle &= U_{\text{good}} V \Pi_0 |S\rangle \qquad (6.22) \\
&= U_{|\psi\rangle} V \Pi_0 |S\rangle,
\end{aligned}$$

or

$$\begin{aligned}
&= U_{|\psi\rangle} |\phi_R\rangle \\
&= U_{|\psi\rangle} \left(\alpha_{00} |\psi, 0\rangle + \alpha_{01} |\psi, 1\rangle + \alpha_{10} \left|\psi^\perp, 0\right\rangle + \alpha_{11} \left|\psi^\perp, 1\right\rangle\right) \qquad (6.23)
\end{aligned}$$

or, if we control the effect of the oracle $U_{|\psi\rangle}$, according to the value of the last register being one,

$$|\phi_A\rangle = \alpha_{00} |\psi, 0\rangle - \alpha_{01} |\psi, 1\rangle + \alpha_{10} \left|\psi^\perp, 0\right\rangle + \alpha_{11} \left|\psi^\perp, 1\right\rangle, \qquad (6.24)$$

with

$$\begin{aligned}
\||\psi_A\rangle\|_2^2 &= \sum_{i,j \in \{0,1\}} \left|(-1)^{j(1-i)} \alpha_{ij}\right|^2 \\
&= \sum_{i,j \in \{0,1\}} \left|\alpha_{ij}\right|^2 \\
&= 1. \qquad (6.25)
\end{aligned}$$

We now observe that

$$|\phi_A\rangle - |\phi_R\rangle = -2\alpha_{01} |\psi, 1\rangle,$$

or

$$|\phi_A\rangle = |\phi_R\rangle - 2\alpha_{01}\,|\psi,1\rangle\,. \tag{6.26}$$

That is, we get

$$\begin{aligned}
\|\Pi_a\,|\phi_A\rangle\|_2^2 &= 1\\
\|\Pi_a\left(|\phi_R\rangle - 2\alpha_{01}\,|\psi,1\rangle\right)\|_2^2 &= 1\\
\left(\sqrt{\mathcal{I}\left(\Pi_a\left(|\phi_R\rangle - 2\alpha_{01}\,|\psi,1\rangle\right),\Pi_a\left(|\phi_R\rangle - 2\alpha_{01}\,|\psi,1\rangle\right)\right)}\right)^2 &= 1\\
\left(\sqrt{\left(\Pi_a\left(|\phi_R\rangle - 2\alpha_{01}\,|\psi,1\rangle\right)\right)^\dagger\Pi_a\left(|\phi_R\rangle - 2\alpha_{01}\,|\psi,1\rangle\right)}\right)^2 &= 1\\
\left(\sqrt{\left(|\phi_R\rangle - 2\alpha_{01}\,|\psi,1\rangle\right)^\dagger\Pi_a^\dagger\Pi_a\left(|\phi_R\rangle - 2\alpha_{01}\,|\psi,1\rangle\right)}\right)^2 &= 1\\
\left(|\phi_R\rangle - 2\alpha_{01}\,|\psi,1\rangle\right)^\dagger\Pi_a^\dagger\Pi_a\left(|\phi_R\rangle - 2\alpha_{01}\,|\psi,1\rangle\right) &= 1\\
\left(|\phi_R\rangle - 2\alpha_{01}\,|\psi,1\rangle\right)^\dagger\Pi_a\Pi_a\left(|\phi_R\rangle - 2\alpha_{01}\,|\psi,1\rangle\right) &= 1\\
\left(|\phi_R\rangle - 2\alpha_{01}\,|\psi,1\rangle\right)^\dagger\Pi_a^2\left(|\phi_R\rangle - 2\alpha_{01}\,|\psi,1\rangle\right) &= 1\\
\left(|\phi_R\rangle - 2\alpha_{01}\,|\psi,1\rangle\right)^\dagger\Pi_a\left(|\phi_R\rangle - 2\alpha_{01}\,|\psi,1\rangle\right) &= 1\\
\left(\langle\phi_R| - 2\alpha_{01}\,\langle\psi,1|\right)\Pi_a\left(|\phi_R\rangle - 2\alpha_{01}\,|\psi,1\rangle\right) &= 1,
\end{aligned} \tag{6.27}$$

or

$$\begin{aligned}
1 = \langle\phi_R|\,\Pi_a\,|\phi_R\rangle &- 2\alpha_{01}\,\langle\phi_R|\,\Pi_a\,|\psi,1\rangle\\
&- 2\alpha_{01}\,\langle\psi,1|\,\Pi_a\,|\phi_R\rangle + 4\alpha_{01}^2\,\langle\psi,1|\,\Pi_a\,|\psi,1\rangle\,.
\end{aligned} \tag{6.28}$$

From this point on, one can make some assumptions about the quantities that emerge in the Equation (6.28), and then proceed with the necessary calculations, that will, hopefully, helps us gain some informative insights about the power of subset states. We decided that is of no good use, for the reader, to provide him, or her, with extra information about what we tried to show, since we are not sure about their soundness whatsoever.

6.3.2 Second Idea

As a second attempt, for $n \in \mathbb{N}$, we tried to sneak uncomputable amplitudes, namely u_i, into the quantum subset-state proof

$$|\psi\rangle = \sum_{i=1}^{2^n} u_i\,|i\rangle\,, \tag{6.29}$$

with $\forall i : u_i \in \mathbb{C}$, and

$$\sum_{i=1}^{2^n} |u_i|^2 = 1, \tag{6.30}$$

in order to separate the class $\mathsf{QMA}_1^{\mathcal{U}}$ not only from the class $\mathsf{SQMA}_1^{\mathcal{U}}$, but from the even more powerful class $\mathsf{QCMA}_{\mathsf{EXP,EXP,1}}^{\mathcal{U}} \supseteq \mathsf{SQMA}_1^{\mathcal{U}}$. Note that the class $\mathsf{QCMA}_{\mathsf{EXP,EXP,1}}^{\mathcal{U}}$ is that version of $\mathsf{QCMA}_{\mathsf{EXP}}^{\mathcal{U}}$ where we have both perfect completeness, and the length of the classical witness-bitstring is exponential, in size, as a function of the input size. Here, the employed oracle, namely $\mathcal{U}$, is the same with that of the Equation (6.13).

Now, let us back up a bit. Why is this approach "wrong?" It is, because sneaking in uncomputable stuff, like complex numbers from the Mandelbrot set [28], surely feels like cheating.

Alright, we now owe to the reader some justifications, about things we wrote earlier. It is the case that

$$\mathsf{QCMA}_{\mathsf{EXP,EXP,1}}^{\mathcal{U}} \supseteq \mathsf{SQMA}_1^{\mathcal{U}}, \tag{6.31}$$

since an exponential-length classical witness can explicitly describe the subset-state quantum proof $|S\rangle$. Also, the quantum machine, which does the verification, is capable of reading the whole witness, since this machine is allotted exponential computation time. We remind that

$$|S\rangle = \sum_{i \in S} \frac{1}{\sqrt{|S|}} |i\rangle\,, \tag{6.32}$$

for some $S \subseteq \left[2^{p(n)}\right]$, where $p\,(n)$ is a polynomial on n, and n is the number of qubits on which our quantum system operates on.

Finally, we can construct a QMA_1-like protocol [13] for the acceptance of the state $|\psi\rangle$, yet it seems that every $\mathsf{QCMA}_{\mathsf{EXP,EXP,1}}$-like protocol will not be able to recognize that $|\psi\rangle$ is a good witness, because even with exponential time, allotted for computation, we cannot compute, or even verify, uncomputable objects.

Note 6.1. We highlight, about the Hadamard transformation, that

$$H\,|0\rangle = \frac{1}{\sqrt{2}}\,|0\rangle + \frac{1}{\sqrt{2}}\,|1\rangle\,, \tag{6.33}$$

and

$$H\,|1\rangle = \frac{1}{\sqrt{2}}\,|0\rangle - \frac{1}{\sqrt{2}}\,|1\rangle\,. \tag{6.34}$$

Also, note the important fact that $H = H^{-1}$, or $H^2 = I$. Thus,

$$H\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) = |0\rangle\,, \tag{6.35}$$

and

$$H\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) = |1\rangle\,. \tag{6.36}$$

The QMA_1 acceptance protocol is this: on input 1^n, for some natural n, and the state $|\psi\rangle$, on n qubits, as the purported proof that $U_n = U_{|\psi\rangle} \in \mathcal{U}_{\text{good}}$, we first prepare the state

$$|0\rangle \otimes |\psi\rangle = |0\rangle\,|\psi\rangle\,, \tag{6.37}$$

and, then, we apply $(H \otimes I^{\otimes n})$ to get the state

$$\begin{aligned}
\left(H \otimes I^{\otimes n}\right)|0\rangle\,|\psi\rangle &= \left(H \otimes I^{\otimes n}\right)|0\rangle \otimes |\psi\rangle \\
&= H\,|0\rangle \otimes I^{\otimes n}\,|\psi\rangle \\
&= \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)|\psi\rangle \\
&= \frac{1}{\sqrt{2}}|0\rangle\,|\psi\rangle + \frac{1}{\sqrt{2}}|1\rangle\,|\psi\rangle \\
&= |\kappa\rangle
\end{aligned} \tag{6.38}$$

or, after we apply the transformation $U_n \in \mathcal{U}$, conditioned on the first qubit being 1, and assuming that $U_n = U_{|\psi\rangle} \in \mathcal{U}_{\text{good}} \subseteq \mathcal{U}$ is true,

$$\begin{aligned}
U_n\,|\kappa\rangle &= \frac{1}{\sqrt{2}}|0\rangle\,|\psi\rangle + \frac{1}{\sqrt{2}}|1\rangle\,U_n\,|\psi\rangle \\
&= \frac{1}{\sqrt{2}}|0\rangle\,|\psi\rangle + \frac{1}{\sqrt{2}}|1\rangle\,(-\,|\psi\rangle) \\
&= \frac{1}{\sqrt{2}}|0\rangle\,|\psi\rangle - \frac{1}{\sqrt{2}}|1\rangle\,|\psi\rangle \\
&= \left|\kappa'\right\rangle.
\end{aligned} \tag{6.39}$$

Now, if we apply $(H \otimes I^{\otimes n})$ to $|\kappa'\rangle$, we have

$$\left(H \otimes I^{\otimes n}\right)\left|\kappa'\right\rangle = \left(H \otimes I^{\otimes n}\right)\left(\frac{1}{\sqrt{2}}|0\rangle\,|\psi\rangle - \frac{1}{\sqrt{2}}|1\rangle\,|\psi\rangle\right)$$

$$
\begin{aligned}
&= \left(H \otimes I^{\otimes n}\right) \left(\left(\frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle \right) |\psi\rangle \right) \\
&= \left(H \otimes I^{\otimes n}\right) \left(\left(\frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle \right) \otimes |\psi\rangle \right) \\
&= H \left(\frac{1}{\sqrt{2}} |0\rangle - \frac{1}{\sqrt{2}} |1\rangle \right) \otimes I^{\otimes n} |\psi\rangle \\
&= |1\rangle \otimes |\psi\rangle \\
&= |1\rangle |\psi\rangle . \qquad (6.40)
\end{aligned}
$$

We observe that a measurement of the first register will yield 1, as a result, with certainty. So, if we measure the first register and get 1, we accept, else we reject. If $U_n \in \mathcal{U}_{\text{good}}$, then we accept with certainty, as we wrote. However, for every 1^m, such that $U_m = I \in \mathcal{U}_{\text{bad}}$, we accept with probability zero. This holds, because in the case where we are handed some $|\phi\rangle$, on $m \in \mathbb{N}$ qubits, to support the false proposition $U_m \in \mathcal{U}_{\text{good}}$, we have, for the, respective to the state $|\kappa'\rangle$, quantum state

$$
|\kappa_2'\rangle = \frac{1}{\sqrt{2}} |0\rangle |\phi\rangle + \frac{1}{\sqrt{2}} |1\rangle |\phi\rangle ,
$$

where $U_m |\phi\rangle = I |\phi\rangle = |\phi\rangle$, that

$$
\begin{aligned}
\left(H \otimes I^{\otimes m}\right) |\kappa_2'\rangle &= \left(H \otimes I^{\otimes m}\right) \left(\frac{1}{\sqrt{2}} |0\rangle |\phi\rangle + \frac{1}{\sqrt{2}} |1\rangle |\phi\rangle \right) \\
&= \left(H \otimes I^{\otimes m}\right) \left(\left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) |\phi\rangle \right) \\
&= \left(H \otimes I^{\otimes m}\right) \left(\left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) \otimes |\phi\rangle \right) \\
&= H \left(\frac{1}{\sqrt{2}} |0\rangle + \frac{1}{\sqrt{2}} |1\rangle \right) \otimes I^{\otimes m} |\phi\rangle \\
&= |0\rangle \otimes |\phi\rangle \\
&= |0\rangle |\phi\rangle . \qquad (6.41)
\end{aligned}
$$

That is, if we measure the first register, we are going to get the result 1 with probability equal to zero! This is clearly a QMA_1 protocol, for this problem.

Bibliography

[1] A. Yu. Kitaev, A. H. Shen, and M. N. Vyalyi. *Classical and Quantum Computation*. Graduate Studies in Mathematics. American Mathematical Society, 2002.

[2] Scott Aaronson. Quantum Computing, Postselection, and Probabilistic Polynomial-Time. *CoRR*, abs/quant-ph/0412187, 2004.

[3] Scott Aaronson. NP-complete Problems and Physical Reality, February 2005.

[4] Scott Aaronson. Oracles Are Subtle But Not Malicious. In *IEEE Conference on Computational Complexity*, pages 340–354. IEEE Computer Society, 2006.

[5] Scott Aaronson. On Perfect Completeness for QMA. *Quantum Information & Computation*, 9(1):81–89, 2009.

[6] Scott Aaronson. BQP and the Polynomial Hierarchy. In Leonard J. Schulman, editor, *STOC*, pages 141–150. ACM, 2010.

[7] Scott Aaronson. Why Philosophers Should Care About Computational Complexity. *Electronic Colloquium on Computational Complexity (ECCC)*, 18:108, 2011.

[8] Scott Aaronson. *Quantum Computing since Democritus*. Cambridge University Press, 2013.

[9] Scott Aaronson, and Andris Ambainis. Forrelation: A Problem that Optimally Separates Quantum from Classical Computing. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *STOC*, pages 307–316. ACM, 2015.

[10] Scott Aaronson, Salman Beigi, Andrew Drucker, Bill Fefferman, and Peter W. Shor. The Power of Unentanglement. *Electronic Colloquium on Computational Complexity (ECCC)*, 15(051), 2008.

[11] Scott Aaronson, and Shalev Ben-David. Sculpting Quantum Speedups. *CoRR*, abs/1512.04016, 2015.

[12] Scott Aaronson, Adam Bouland, Joseph Fitzsimons, and Mitchell Lee. The Space "Just Above" BQP. In Madhu Sudan, editor, *ITCS*, pages 271–280. ACM, 2016.

[13] Scott Aaronson, and Greg Kuperberg. Quantum Versus Classical Proofs and Advice. *Electronic Colloquium on Computational Complexity (ECCC)*, 13(055), 2006.

[14] L. Adleman, J. Demarrais, and M.-D. Huang. Quantum Computability. *SIAM Journal of Computation*, pages 1524–1540, 1997.

[15] Dorit Aharonov. Quantum Computation, 1998.

[16] Dorit Aharonov, Itai Arad, and Thomas Vidick. The Quantum PCP Conjecture. *CoRR*, abs/1309.7495, 2013.

[17] Dorit Aharonov and Tomer Naveh. Quantum NP—A Survey. 2002.

[18] Dorit Aharonov, Wim van Dam, Julia Kempe, Zeph Landau, Seth Lloyd, and Oded Regev. Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation. In *FOCS*, pages 42–51. IEEE Computer Society, 2004.

[19] Andris Ambainis. Quantum Lower Bounds by Quantum Arguments. *Journal of Computer and System Sciences*, 64, 2002.

[20] László Babai. Trading Group Theory for Randomness. In Robert Sedgewick, editor, *STOC*, pages 421–429. ACM, 1985.

[21] László Babai. Graph Isomorphism in Quasipolynomial Time. *CoRR*, abs/1512.03547, 2015.

[22] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum Lower Bounds by Polynomials. *J. ACM*, 48(4):778–797, July 2001.

[23] J. S. Bell. On the Einstein-Podolsky-Rosen Paradox. *Physics*, 1:195–200, 1964.

[24] Charles Bennett. Logical Reversibility of Computation. *IBM J. Res. Dev.*, 17(6):525–532, November 1973.

[25] Charles Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and Weaknesses of Quantum Computing. *SIAM J. Comput.*, 26(5):1510–1523, October 1997.

[26] André Berthiaume, and Gilles Brassard. Oracle Quantum Computing. *Journal of Modern Optics*, 41(12):2521–2535, 1994.

[27] Bill Fefferman, and Shelby Kimmel. Quantum vs Classical Proofs and Subset Verification. 2015.

[28] Lenore Blum, Felipe Cucker, Michael Shub, and Steve Smale. *Complexity and Real Computation*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1998.

[29] Cris Calude, and Gheorghe Paun. *Computing with Cells and Atoms: An Introduction to Quantum, DNA and Membrane Computing*. CRC Press, First edition, 2000.

[30] Richard Chang, Benny Chor, Oded Goldreich, Juris Hartmanis, Johan Hastad, Desh Ranjan, and Pankaj Rohatgi. The Random Oracle Hypothesis is False. *J. Comput. Syst. Sci.*, 49(1):24–39, August 1994.

[31] Charles Bennett, and John Gill. Relative to a Random Oracle $\mathcal{A}$, $\mathsf{P}^{\mathcal{A}} \neq \mathsf{NP}^{\mathcal{A}} \neq \mathsf{coNP}^{\mathcal{A}}$ with Probability 1. *SIAM Journal on Computing*, 10, 1981.

[32] Lijie Chen. A Note on Oracle Separations for BQP. *CoRR*, abs/1605.00619, 2016.

[33] Stephen A. Cook. The Complexity of Theorem-Proving Procedures. 1971.

[34] Cristopher Moore, and Stephan Mertens. *The Nature of Computation*. Oxford University Press, USA, 2011.

[35] J. Niel de Beaudrap, Richard Cleve, and John Watrous. Sharp Quantum versus Classical Query Complexity Separations, 2002.

[36] David Deutsch. *Quantum Theory, The Church-Turing Principle & the Universal Quantum Computer*.

[37] David Deutsch. Quantum Computational Networks. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 425(1868):73–90, 1989.

[38] David Deutsch. *The Fabric of Reality*. Penguin (Non-Classics), 1998.

[39] David Deutsch. Quantum Theory of Probability and Decisions. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 455(1988):3129–3137, 1999.

[40] Donald Knuth, Tracy Larrabee, and Paul Roberts. *Mathematical Writing*. Cambridge University Press, 1989.

[41] A. Einstein, B. Podolsky, and N. Rosen. Can Quantum-Mechanical Description of Physical Reality Be Considered Complete? *Phys. Rev.*, 47(10):777–780, May 1935.

[42] Eleanor Rieffel, and Wolfgang Polak. *Quantum Computing: A Gentle Introduction*. Scientific and Engineering Computation. The MIT Press, 2011.

[43] Ethan Bernstein, and Umesh Vazirani. Quantum Complexity Theory. In *Proc. 25th Annual ACM Symposium on Theory of Computing, ACM*, pages 11–20, 1993.

[44] Stephen Fenner, Lance Fortnow, Stuart Kurtz, and Lide Li. An Oracle Builder's Toolkit. *Inf. Comput.*, 182(2):95–136, May 2003.

[45] Richard P. Feynman. Simulating Physics with Computers. *International Journal of Theoretical Physics*, 21(6):467–488, 1982.

[46] Richard P. Feynman. Quantum Mechanical Computers. *Optics News*, 11, 1985.

[47] Lance Fortnow. One Complexity Theorist's View of Quantum Computing. *Theor. Comput. Sci.*, 292(3):597–610, January 2003.

[48] Fortnow, L., and Rogers, J. Complexity Limitations on Quantum Computation. In *Proceedings of the Thirteenth Annual IEEE Conference on Computational Complexity*, COCO '98, Washington, DC, USA, 1998. IEEE Computer Society.

[49] Michael R. Garey, and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.

[50] Sevag Gharibian, Yichen Huang, Zeph Landau, and Seung Woo Shin. Quantum Hamiltonian Complexity. *Foundations and Trends in Theoretical Computer Science*, 10(3):159–282, 2015.

[51] Kurt Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte fr mathematik und physik*, 38(1):173–198, 1931.

[52] Oded Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2008.

[53] Ronald Graham, Donald Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley, Reading, 1989.

[54] Robert Griffiths. *Consistent Quantum Theory*. Cambridge University Press, 2002.

[55] Alex Bredariol Grilo, Iordanis Kerenidis, and Jamie Sikora. *QMA with Subset State Witnesses*, pages 163–174. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[56] P. R. Halmos. *Naive Set Theory*. Undergraduate Texts in Mathematics. Springer, 1974.

[57] Yenjo Han, Lane Hemaspaandra, and Thomas Thierauf. Threshold Computation and Cryptographic Security. *SIAM J. Comput.*, 26(1):59–78, 1997.

[58] Harry Lewis, and Christos Papadimitriou. *Elements of the Theory of Computation*. Prentice-Hall, 1998.

[59] Rahul Jain, Zhengfeng Ji, Sarvagya Upadhyay, and John Watrous. $\mathsf{QIP} = \mathsf{PSPACE}$. *J. ACM*, 58(6):30, 2011.

[60] James Brown, and Ruel Churchill. *Complex Variables and Applications*. McGraw-Hill Education, Ninth edition, 2013.

[61] John Hopcroft, and Jeffrey Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley Series in Computer Science. Addison-Wesley Publishing Company, 1979.

[62] Jon Kleinberg, and Éva Tardos. *Algorithm Design*. Addison Wesley, 2005.

[63] Stephen P. Jordan, Hirotada Kobayashi, Daniel Nagaj, and Harumichi Nishimura. Achieving Perfect Completeness in Classical-witness Quantum Merlin-Arthur Proof Systems. *Quantum Info. Comput.*, 12(5-6):461–471, May 2012.

[64] Richard M. Karp. *Reducibility Among Combinatorial Problems*, pages 85–103. Springer US, Boston, MA, 1972.

[65] A. Yu. Kitaev. Quantum Computations: Algorithms and Error Correction. *Russian Mathematical Surveys*, 52, 1997.

[66] E. Knill. Quantum Randomness and Non-Determinism. 1996.

[67] Dexter Kozen. *Automata and Computability*. Undergraduate Texts in Computer Science. Springer, 1997.

[68] Lane Hemaspaandra, and Mitsunori Ogihara. *The Complexity Theory Companion*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2001.

[69] Leonard Susskind, and Art Friedman. *Quantum Mechanics: The Theoretical Minimum*. Basic Books, 2014.

[70] Seth Lloyd. Universe as Quantum Computer. *Complexity*, 3(1):32–35, 1997.

[71] Seth Lloyd. Complexity: Plain and Simple. *Complexity*, 4(4):72, 1999.

[72] Seth Lloyd. Quantum Procrastination. *Science*, 338(6107):621–622, November 2012.

[73] Seth Lloyd. A Turing Test for Free Will. *CoRR*, abs/1310.3225, 2013.

[74] Chris Marriott, and John Watrous. Quantum Arthur-Merlin Games. *Computational Complexity*, 14:2005.

[75] David McMahon. *Quantum Mechanics Demystified*. McGraw-Hill Professional, 2005.

[76] David McMahon. *Quantum Computing Explained*. Wiley-Interscience, IEEE Computer Society, 2008.

[77] Michael A. Nielsen, and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge Series on Information and the Natural Sciences. Cambridge University Press, 2004.

[78] Murray Spiegel, R. Alu Srinivasan, and John Schiller. *Schaum's Outline of Probability and Statistics*. Schaum's Outline Series. McGraw-Hill, 2008.

[79] Noson Yanofsky, and Mirco Mannucci. *Quantum Computing for Computer Scientists*. Cambridge University Press, 2008.

[80] Christos Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[81] Paul G. Hoel, Sidney C. Port, and Charles J. Stone. *Introduction to Probability Theory*. The Houghton-Mifflin Series in Statistics. Houghton Mifflin, First edition, 1971.

[82] Roger Penrose, and C. J. Isham. *Quantum Concepts in Space and Time*. New York, Oxford University Press, 1986.

[83] Phillip Kaye, Raymond Laflamme, and Michele Mosca. *An Introduction to Quantum Computing*. Oxford University Press, USA, 2007.

[84] Arthur Pittenger. *An Introduction to Quantum Computing Algorithms*. Progress in Computer Science and Applied Logic 19. Birkhäuser Basel, 2000.

[85] Hartley Rogers. *Theory of Recursive Functions and Effective Computability*. The MIT Press, 1987.

[86] Kenneth Rosen. *Discrete Mathematics and Its Applications*. McGraw Hill Higher Education, 2006.

[87] Sanjeev Arora, and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[88] Sanjoy Dasgupta, Christos Papadimitriou, and Umesh Vazirani. *Algorithms*. McGraw-Hill Science/Engineering/Math, 2006.

[89] Uwe Schning, and Randall Pruim. *Gems of Theoretical Computer Science*. Springer, 1998.

[90] Scott Aaronson, and John Watrous. Closed Timelike Curves Make Quantum and Classical Computing Equivalent. *Mathematical Physical & Engineering Sciences*, 465, 2009.

[91] Seymour Lipschutz, and Marc Lipson. *Schaum's Outline of Linear Algebra*. Schaum's Outline. Fourth edition, 2009.

[92] Peter W. Shor. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In *FOCS*, pages 124–134. IEEE Computer Society, 1994.

[93] Daniel R. Simon. On the Power of Quantum Computation. *SIAM Journal on Computing*, 26:116–123, 1994.

[94] Michael Sipser. *Introduction to the Theory of Computation*. Thomson Course Technology, 2006.

[95] Willi-Hans Steeb, and Yorick Hardy. *Problems and Solutions in Quantum Computing and Quantum Information*. World Scientific Publishing Company, 2004.

[96] T. Baker, J. Gill, and R. Solovay. Relativizations of the $\mathsf{P} \overset{?}{=} \mathsf{NP}$ Question. *SIAM Journal on Computing*, 4, December 1975.

[97] Thomas Cormen, Charles Leiserson, Ronald Rivest, and Clifford Stein. *Introduction to Algorithms*. McGraw-Hill Science/Engineering/Math, 2001.

[98] Timothy Gowers, June Barrow-Green, and Imre Leader. *The Princeton Companion to Mathematics*. Princeton University Press, 2008.

[99] S. Toda. [IEEE Comput. Soc. Press [1990] 31st Annual Symposium on Foundations of Computer Science - St. Louis, MO, USA (22-24 Oct. 1990)] Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science - The Complexity of Finding Medians. 1990.

[100] S. Toda. PP is As Hard As the Polynomial-time Hierarchy. *SIAM J. Comput.*, 20(5):865–877, October 1991.

[101] Alan M. Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42, 1937.

[102] Alan M. Turing. Computing Machinery and Intelligence. *Mind*, 59:433–460, 1950.

[103] John Watrous. Quantum Computational Complexity. In Robert A. Meyers, editor, *Encyclopedia of Complexity and Systems Science*, pages 7174–7201. Springer, 2009.

[104] Dominic Welsh. *Complexity: Knots, Colourings and Counting*. Cambridge University Press, 1993.

[105] Peter Wittek. *Quantum Machine Learning: What Quantum Computing Means to Data Mining*. Elsevier Insights. Elsevier AP, Academic Press, 2014.

[106] Stathis Zachos, and Martin Fürer. Probabilistic Quantifiers vs. Distrustful Adversaries. In *Proc. of the Seventh Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 443–455, London, UK, 1987. Springer-Verlag.

Kindly note that, during the preparation process of this thesis, there were consumed—by the author, of course—about seventy-four cups of espresso, thirty potato burgers, twenty-two litres of almond milk, and a gigantic, able to talk, marshmallow.

Please report any side effects. □