Quantum Random Access Codes with Shared Randomness

by

Maris Ozols

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2009

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

We consider a communication method, where the sender encodes n classical bits into 1 qubit and sends it to the receiver who performs a certain measurement depending on which of the initial bits must be recovered. This procedure is called $n \overset{p}{\mapsto} 1$ quantum random access code (QRAC) where $p > 1/2$ is its success probability. It is known that $2 \xmapsto{0.85} 1$ and $3 \xmapsto{0.79} 1$ QRACs (with no classical counterparts) exist and that $4 \overset{p}{\mapsto} 1$ QRAC with $p > 1/2$ is not possible.

We extend this model with shared randomness (SR) that is accessible to both parties. Then $n \overset{p}{\mapsto} 1$ QRAC with SR and $p > 1/2$ exists for any $n \geq 1$. We give an upper bound on its success probability (the known $2 \xmapsto{0.85} 1$ and $3 \xmapsto{0.79} 1$ QRACs match this upper bound). We discuss some particular constructions for several small values of n.

We also study the classical counterpart of this model where n bits are encoded into 1 bit instead of 1 qubit and SR is used. We give an optimal construction for such codes and find their success probability exactly—it is less than in the quantum case.

Interactive 3D quantum random access codes are available on-line at
`http://home.lanet.lv/∼sd20008/racs`

Acknowledgements

Most of these results were obtained in the summer of 2006 during the undergraduate summer research program at the University of Waterloo. I worked together with Laura Mancinska under the supervision of Andris Ambainis and Debbie Leung.

I would like to thank Laura, because none of this would be possible without her. I would also like to thank our supervisors Andris and Debbie for believing in us and giving a place for both of us in the summer research program. It was really a pleasure to work with both of you! I would also like to thank the University of Waterloo and IQC for hospitality.

I received many valuable comments from the readers of this thesis: Debbie Leung, Andrew Childs, and Ashwin Nayak. I appreciate your feedback.

Finally, I would like to thank my present supervisor Andrew Childs for always being open to discuss research.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Random access codes

In general *random access code* (or simply RAC) stands for "encoding a long message into fewer bits with the ability to recover (decode) any one of the initial bits (with some probability of success)". A random access code can be characterized by the symbol "$n \overset{p}{\mapsto} m$" meaning that n bits are encoded into m and any one of the initial bits can be recovered with probability at least p. We require that $p > 1/2$ since $p = 1/2$ can be achieved by guessing. In this paper we consider only the case when $m = 1$. So we have the following problem:

Problem (Classical)**.** *There are two parties—Alice and Bob. Alice is asked to encode some classical n-bit string into* 1 *bit and send this bit to Bob. We want Bob to be able to recover any one of the n initial bits with high success probability.*

Note that Alice does not know in advance which bit Bob will need to recover, so she cannot send only that bit. If they share a quantum channel then we have the quantum version of the previous problem:

Problem (Quantum)**.** *Alice must encode her classical n-bit message into* 1 qubit *(quantum bit) and send it to Bob. He performs some measurement on the received qubit to extract the required bit (the measurement that is used depends on which bit is needed).*

Both problems look similar, however the quantum version has an important feature. In the classical case the fact that Bob can recover any one of the initial bits implies that he can actually recover all of them—each with high probability of success. Surprisingly in the quantum case this is not true, because after the first measurement the state of the qubit will be disturbed and further attempts to extract more information can fail.

1.2 History and applications

As noted in [6, 8], the idea behind *quantum random access codes* or QRACs is very old (relative to quantum information standards). It first appeared in a paper by

Stephen Wiesner [1] published in 1983 and was called *conjugate coding*. Later these codes were re-discovered by Ambainis et al. in [2, 3]. They show that there exists $2 \xrightarrow{0.85} 1$ QRAC and mention its immediate generalization to $3 \xrightarrow{0.79} 1$ QRAC due to Chuang (see also [5] and [8] for more details). However, Hayashi et al. [5] show that it is impossible to construct a $4 \overset{p}{\mapsto} 1$ QRAC with $p > 1/2$. We will discuss these results more in Sect. 3.3.

There has also been work on $n \overset{p}{\mapsto} m$ codes with $m > 1$, see [2, 3, 4]. Ambainis et al. [2] show that if a $n \overset{p}{\mapsto} m$ QRAC with $p > 1/2$ exists, then $m = \Omega(n/\log n)$, which was later improved by Nayak [4, 3] to $m \geq (1 - H(p))n$, where $H(p) = -p \log p - (1-p)\log(1-p)$ is the *binary entropy function*. Other generalizations include: considering d-valued bits instead of qubits [6, 8] and recovering several rather than a single bit [16].

Originally quantum random access codes were studied in the context of quantum finite automata [2, 3, 4]. However, they also have applications in quantum communication complexity [6, 9, 10, 11], in particular for *network coding* [5, 12] and *locally decodable codes* [13, 14, 15, 16]. Recently results on quantum random access codes have been applied for quantum state learning [17].

Experimental feasibility of QRACs and their relation to *contextuality* and *non-locality* has been discussed in [6, Chapter 7]. Recently a similar protocol called *parity-oblivious multiplexing* has been considered in [7]. It has an additional cryptographic constraint that Alice is not allowed to transmit any information about the parity of the input string. In addition [7] also discuss the first experimental demonstration of $2 \mapsto 1$ and $3 \mapsto 1$ QRACs.

We want to emphasize the setting in which the impossibility of $4 \overset{p}{\mapsto} 1$ QRAC with $p > 1/2$ was proved in [5]: Alice is allowed to perform a locally randomized encoding of the given string into a one-qubit state and Bob is allowed to perform different *positive operator-valued measure* (POVM) measurements to recover different bits. This is the most general setting when information is encoded into a one-qubit state and both parties are allowed to use randomized strategies, but only have access to local coins. However, we can consider an even more general setting—when both parties share a common coin. This means that Alice and Bob are allowed to cooperate by using some shared source of randomness to agree on which strategy to use. We will refer to this source as a *shared random string* or *shared randomness* (SR). Note that shared randomness is a more powerful resource than local randomness, since parts of the shared random string can be exclusively used only by Alice or Bob to simulate local coins. It turns out that in this new setting $4 \overset{p}{\mapsto} 1$ QRAC is possible with $p > 1/2$. In fact, $n \overset{p}{\mapsto} 1$ QRACs with $p > 1/2$ can be constructed for all $n \geq 1$ (see Sect. 3.8).

1.3 Outline of results

In Sect. 2 we study classical $n \mapsto 1$ random access codes with shared randomness. In Sect. 2.2 we introduce Yao's principle that is useful for understanding both classical and quantum codes. A classical code that is optimal for all n is presented in Sect. 2.3.1 and the asymptotic behavior of its success probability is considered in Sect. 2.3.2.

In Sect. 3 we study quantum random access codes with shared randomness. In Sect. 3.3 we discuss what is known in the case when shared randomness is not allowed, i.e., $2 \mapsto 1$ and $3 \mapsto 1$ QRACs and the impossibility of $4 \mapsto 1$ QRAC. In Sect. 3.6 we give an upper bound of success probability of QRACs with SR and generalize it in Sect. 3.7 for POVM measurements. In Sect. 3.8 we give two constructions of $n \overset{p}{\mapsto} 1$ QRAC with SR and $p > 1/2$ for all $n \geq 2$ that provide a lower bound for success probability.

In Sect. 4 we try to find optimal QRACs with SR for several small values of n. In particular, in Sect. 4.1 we discuss QRACs obtained by numerical optimization, and in Sect. 4.2 we consider symmetric constructions.

Finally, we conclude in Sect. 5 with a summary of the obtained results (Sect. 5.1), a list of open problems (Sect. 5.2) and possible generalizations (Sect. 5.3).

Chapter 2

Classical random access codes

2.1 Types of classical encoding-decoding strategies

As a synonym for random access code we will use the term *strategy* to refer to the joint *encoding-decoding scheme* used by Alice and Bob. Two measures of how good the strategy is will be used: the *worst case success probability* and the *average success probability.* Both probabilities must be calculated over all possible pairs (x, i) where $x \in \{0,1\}^n$ is the input and $i \in \{1, \dots, n\}$ indicates which bit must be recovered. We are interested in the worst case success probability, but in our case according to Yao's principle (introduced in Sect. 2.2) the average success probability can be used to estimate it.

Depending on the computational model considered, different types of strategies are allowed. The simplest type corresponds to Alice and Bob acting deterministically and independently.

Definition. A *pure classical* $n \mapsto 1$ *encoding-decoding strategy* is an ordered tuple $(E, D_1, \dots, D_n)$ that consists of an *encoding function* $E : \{0,1\}^n \mapsto \{0,1\}$ and n *decoding functions* $D_i : \{0,1\} \mapsto \{0,1\}$.

These limited strategies yield RACs with poor performance. This is because Bob can recover all bits correctly for no more than two input strings, since he receives either 0 or 1 and acts deterministically in each case. For all other strings at least one bit will definitely be recovered incorrectly, therefore the worst case success probability is 0. If we allow Alice and Bob to act probabilistically but without cooperation, then we get mixed strategies.

Definition. A *mixed classical* $n \mapsto 1$ *encoding-decoding strategy* is an ordered tuple $(P_E, P_{D_1}, \dots, P_{D_n})$ of probability distributions. P_E is a distribution over encoding functions and P_{D_i} over decoding functions.

It is obvious that in this setting the worst case probability is at least 1/2. This is obtained by guessing—we output either 0 or 1 with probability 1/2 regardless of the input. Formally this means that for each i, P_{D_i} is a uniform distribution over two constant decoding functions 0 and 1. It has been shown that in this setting for $2 \mapsto 1$ case one cannot do better than guessing, i.e., there is no $2 \overset{p}{\mapsto} 1$ RAC with worst case success probability $p > 1/2$ [2, 3].

However, we can allow cooperation between Alice and Bob—they can use a shared random string to agree on some joint strategy.

Definition. A *classical $n \mapsto 1$ encoding-decoding strategy with shared randomness* is a probability distribution over pure classical strategies.

Note that this is the most general randomized setting, since both randomized cooperation and local randomization are possible. This is demonstrated in the following example.

Example. Consider the following strategy: randomly agree on $i \in \{1, \dots, n\}$ and send the ith bit; if the ith bit is requested, output the received bit, otherwise guess. This strategy can formally be specified as follows: uniformly choose a pure strategy from the set

$$\bigcup_{i \in \{1,\dots,n\}} \left\{ (e_i, c_1, \dots, c_{i-1}, d, g_1, \dots, g_{n-i}) \mid c \in \{d_0, d_1\}^{i-1}, g \in \{d_0, d_1\}^{n-i} \right\}, \quad (2.1)$$

where the encoding function e_i is given by $e_i(x) = x_i$ and decoding functions d_0, d_1, and d are given by $d_0(b) = 0$, $d_1(b) = 1$, and $d(b) = b$, where b is the received bit. The total amount of required randomness is $n - 1 + \log n$ bits, because one out of $n \cdot 2^{n-1}$ pure strategies must be selected. Note that only $\log n$ of these bits must be shared among Alice and Bob, so that they can agree on the value of i. The remaining $n-1$ random bits are needed only by Bob for choosing random decoding functions $c \in \{d_0, d_1\}^{i-1}$ and $g \in \{d_0, d_1\}^{n-i}$.

Note that the amount of randomness used in the above example can be reduced. Since only one bit must be recovered, there is no need to choose each of the decoding functions independently. Thus Bob needs only one random bit that he will output whenever some bit other than the ith bit is requested. This is illustrated in the next example.

Example. Alice and Bob uniformly sample a pure strategy from the following set:

$$\left\{ (e_i, \underbrace{c, \dots, c}_{i-1}, d, \underbrace{c, \dots, c}_{n-i}) \mid 1 \leq i \leq n, c \in \{d_0, d_1\} \right\}. \quad (2.2)$$

This requires $\log n$ random bits to be shared among Alice and Bob and 1 private random bit for Bob, i.e., $1 + \log n$ random bits in total.

We are interested in classical strategies with SR, because they provide a classical analogue of QRACs with SR. However, in this setting finding the optimal strategy seems to be hard, therefore we will turn to Yao's principle for help.

2.2 Yao's principle

When dealing with randomized algorithms, it is hard to draw general conclusions (like proving optimality of a certain randomized algorithm) because the possible algorithms may form a continuum. In such situations it is very helpful to apply Yao's principle [18]. This allows us to shift the randomness in the algorithm to the input and consider only deterministic algorithms.

Let $\mathcal{S}$ be a classical strategy with SR. One can think of it as a stochastic process consisting of applying the encoding map E to the input x, followed by applying the decoding map D_i to the ith bit. Both of these maps depend on the value of the shared random string. The result of $\mathcal{S}$ is $\mathcal{S}(x,i) = D_i(E(x))$, which is a stochastic variable over the set $\{0,1\}$. Let $\Pr[\mathcal{S}(x,i) = x_i]$ denote the probability that the stochastic variable $\mathcal{S}(x,i)$ takes value x_i. Then the worst case success probability of the optimal classical strategy with SR is given by

$$\max_{\mathcal{S}} \min_{x,i} \Pr[\mathcal{S}(x,i) = x_i]. \tag{2.3}$$

Let μ be some distribution over the input set $\{0,1\}^n \times \{1,\ldots,n\}$ and let $\Pr_\mu[\mathcal{P}(x,i) = x_i]$ denote the expected success probability of a pure (deterministic) strategy $\mathcal{P}$. If the "hardest" input distribution is chosen as μ, then the expected success probability of the best pure strategy for this distribution is

$$\min_{\mu} \max_{\mathcal{P}} \Pr_\mu[\mathcal{P}(x,i) = x_i]. \tag{2.4}$$

Yao's principle states that the quantities given in (2.3) and (2.4) are equal [18]:

$$\max_{\mathcal{S}} \min_{x,i} \Pr[\mathcal{S}(x,i) = x_i] = \min_{\mu} \max_{\mathcal{P}} \Pr_\mu[\mathcal{P}(x,i) = x_i]. \tag{2.5}$$

Thus Yao's principle provides us with an upper bound for the worst case probability (2.3). All we have to do is to choose an arbitrary input distribution μ_0 and find the best pure strategy $\mathcal{P}_0$ for it. Then according to Yao's principle we have

$$\Pr_{\mu_0}[\mathcal{P}_0(x,i) = x_i] \geq \max_{\mathcal{S}} \min_{x,i} \Pr[\mathcal{S}(x,i) = x_i], \tag{2.6}$$

with equality if and only if μ_0 is the "hardest" distribution. It turns out that for random access codes the uniform distribution η is the "hardest". To prove it, we must first consider the randomization lemma.

Lemma 1. $\forall \mathcal{P} \exists \mathcal{S} : \min_{x,i} \Pr[\mathcal{S}(x,i) = x_i] = \Pr_\eta[\mathcal{P}(x,i) = x_i]$, *where η is the uniform distribution. In other words: the worst case success probability of $\mathcal{S}$ is the same as the average case success probability of $\mathcal{P}$ with uniformly distributed input.*

Proof. This can be achieved by randomizing the input with the help of the shared random string. Alice's input can be randomized by XOR-ing it with an n-bit random string r. But Bob's input can be randomized by adding (modulo n) a random number $d \in \{0,\ldots,n-1\}$ to it (assume for now that bits are numbered from 0 to $n-1$). To obtain a consistent strategy, these actions must be identically performed on both sides, thus a shared random string of $n + \log n$ bits[1] is required. Assume that E and D_i are the encoding and decoding functions of the pure strategy $\mathcal{P}$; then the new strategy $\mathcal{S}$ is

$$E'(x) = E(\mathrm{Shift}_d(x \oplus r)), \tag{2.7}$$

$$D'_i(b) = D_{i+d \bmod n}(b) \oplus r_i, \tag{2.8}$$

where $\mathrm{Shift}_d(s)$ substitutes $s_{i+d \bmod n}$ by s_i in string s. Due to input randomization, this strategy has the same success probability for all inputs (x, i), namely

$$\Pr[\mathcal{S}(x,i) = x_i] \;=\; \sum_{y \in \{0,1\}^n} \sum_{j=0}^{n-1} \frac{1}{2^n \cdot n} \Pr[\mathcal{P}(y,j) = y_j] = \Pr_\eta[\mathcal{P}(y,j) = y_j], \tag{2.9}$$

coinciding with the average success probability of the pure strategy $\mathcal{P}$. □

Now we will show that inequality (2.6) becomes an equality when $\mu_0 = \eta$, meaning that the uniform distribution η is the "hardest".

Lemma 2. *The minimum of (2.4) is reached at the uniform distribution η, i.e.,*

$$\min_\mu \max_{\mathcal{P}} \Pr_\mu[\mathcal{P}(x,i) = x_i] = \max_{\mathcal{P}} \Pr_\eta[\mathcal{P}(x,i) = x_i]. \tag{2.10}$$

Proof. From the previous Lemma we know that there exists a strategy with SR $\mathcal{S}_0$ such that

$$\min_{x,i} \Pr[\mathcal{S}_0(x,i) = x_i] = \max_{\mathcal{P}} \Pr_\eta[\mathcal{P}(x,i) = x_i] \tag{2.11}$$

($\mathcal{S}_0$ is obtained from the best pure strategy by prepending it with input randomization). However, among all strategies with SR there might be one that is better than $\mathcal{S}_0$, thus

$$\max_{\mathcal{S}} \min_{x,i} \Pr[\mathcal{S}(x,i) = x_i] \geq \max_{\mathcal{P}} \Pr_\eta[\mathcal{P}(x,i) = x_i]. \tag{2.12}$$

But if we put $\mu_0 = \eta$ into inequality (2.6), we obtain

$$\max_{\mathcal{P}} \Pr_\eta[\mathcal{P}(x,i) = x_i] \geq \max_{\mathcal{S}} \min_{x,i} \Pr[\mathcal{S}(x,i) = x_i], \tag{2.13}$$

which is the same as (2.12), but with reversed sign. This means that both sides are actually equal:

$$\max_{\mathcal{P}} \Pr_\eta[\mathcal{P}(x,i) = x_i] = \max_{\mathcal{S}} \min_{x,i} \Pr[\mathcal{S}(x,i) = x_i]. \tag{2.14}$$

Applying Yao's principle to the right hand side of (2.14) we obtain the desired equation (2.10). □

[1] We will not worry about how Bob obtains a uniformly distributed d from a string of random bits when $n \neq 2^k$.

Theorem 1. *For any pure strategy $\mathcal{P}$*

$$\Pr_\eta[\mathcal{P}(x,i) = x_i] \leq \max_{\mathcal{S}} \min_{x,i} \Pr[\mathcal{S}(x,i) = x_i], \tag{2.15}$$

with equality if and only if $\mathcal{P}$ is optimal for the uniform distribution η.

Proof. To obtain the required inequality, do not maximize the left hand side of equation (2.14), but put an arbitrary $\mathcal{P}$. It is obvious that we will obtain equality if and only if $\mathcal{P}$ is optimal. □

This theorem has important consequences—it allows us to consider pure strategies with uniformly distributed input rather than strategies with SR. If we manage to find the optimal pure strategy, then we can also construct an optimal strategy with SR using input randomization[2]. If the pure strategy is not optimal, then we get a lower bound for the strategy with SR.

2.3 Classical $n \mapsto 1$ RAC

Before considering $n \mapsto 1$ QRACs with shared randomness, we will find an optimal classical $n \mapsto 1$ RAC with shared randomness and derive bounds for it.

2.3.1 Optimal strategy

According to Theorem 1 we can consider only pure strategies. As a pure strategy is deterministic, for each input it gives either a correct or a wrong answer. To maximize the average success probability we must find a pure strategy that gives the correct answer for as many of the $n \cdot 2^n$ inputs as possible—such a strategy we will call an *optimal pure strategy.*

Let us first consider the problem of finding an optimal decoding strategy, when the encoding strategy is fixed. An encoding function $E : \{0,1\}^n \mapsto \{0,1\}$ divides the set of all strings into two parts:

$$\begin{aligned} X_0 &= \{x \in \{0,1\}^n \mid E(x) = 0\}, \\ X_1 &= \{x \in \{0,1\}^n \mid E(x) = 1\}. \end{aligned} \tag{2.16}$$

If Bob receives bit b, he knows that the initial string was definitely from the set X_b, but there is no way for him to tell exactly which string it was. However, if he must recover only the ith bit, he can check whether there are more zeros or ones among the ith bits of strings from set X_b. More formally, we can introduce the symbol $N_i^b(k)$ that denotes the number of strings from set X_b that have the bit k in ith position:

$$N_i^b(k) = |\{x \in X_b \mid x_i = k\}|, \tag{2.17}$$

[2] If the encoding function depends only on the Hamming weight of the input string x (e.g., majority function) and the decoding function does not depend on i, there is no need to randomize over i, so n instead of $n + \log n$ shared random bits are enough.

Therefore the optimal decoding strategy $D_i : \{0,1\} \mapsto \{0,1\}$ for the ith bit is

$$D_i(b) = \begin{cases} 0 & \text{if } N_i^b(0) \geq N_i^b(1), \\ 1 & \text{otherwise.} \end{cases} \tag{2.18}$$

Of course, if $N_i^b(0) = N_i^b(1)$, Bob can output 1 as well. For pure strategies there are only 4 possible decoding functions for each bit: 0, 1, b, or NOT b. But this is still quite a lot so we will consider the two following lemmas. The first lemma will rule out the *constant decoding functions* 0 and 1.

Lemma 3. *For any n there exists an optimal pure classical $n \mapsto 1$ RAC that does not use constant decoding functions* 0 *and* 1 *for any bits.*

Proof. We will show that if there exists an optimal strategy that contains constant decoding functions for some bits, then there also exists an optimal strategy that does not. Let us assume that there is an optimal strategy with constant decoding function 0 for the ith bit (the same argument goes through for 1 as well). Then according to equation (2.18) we have $N_i^0(0) \geq N_i^0(1)$ and $N_i^1(0) \geq N_i^1(1)$. Note that $N_i^0(0) + N_i^1(0) = N_i^0(1) + N_i^1(1) = 2^{n-1}$, because $x_i = 0$ in exactly half of all 2^n strings. This means that actually $N_i^0(0) = N_i^0(1)$ and $N_i^1(0) = N_i^1(1)$. If we take a look at (2.18) again, we see that in such situation any decoding strategy is optimal and we can use any non-constant strategy instead. □

Lemma 4. *For any n there exists an optimal pure classical $n \mapsto 1$ RAC that does not use decoding function* NOT b *for any bits.*

Proof. We will show that for each pure strategy $\mathcal{P}$ that uses negation as the decoding function for the ith bit, there exists a pure strategy $\mathcal{P}'$ with the same average case success probability that does not. If $\mathcal{P}$ consists of encoding function E and decoding functions D_j, then $\mathcal{P}'$ can be obtained from $\mathcal{P}$ by inverting the ith bit before encoding and after decoding:

$$E'(x) = E(\mathrm{NOT}_i\, x), \tag{2.19}$$

$$D'_j(b) = \begin{cases} \mathrm{NOT}\, D_j(b) & \text{if } j = i, \\ D_j(b) & \text{otherwise,} \end{cases} \tag{2.20}$$

where NOT_i inverts the ith bit of string. It is obvious that $\mathcal{P}$ and $\mathcal{P}'$ have the same average success probabilities, because if $\mathcal{P}$ gives the correct answer for input (x, i) then $\mathcal{P}'$ gives the correct answer for input $(\mathrm{NOT}_i\, x, i)$. The same holds for wrong answers. □

Theorem 2. *The pure classical $n \mapsto 1$ RAC with identity decoding functions and majority encoding function is optimal.*

Proof. According to Lemma 3 and Lemma 4, there exists an optimal pure classical $n \mapsto 1$ RAC with identity decoding function for all bits. Now we must consider the other part—finding an optimal encoding given a particular (identity) decoding function. It is obvious that in our case optimal encoding must return the majority of bits:

$$E'(x) = \begin{cases} 0 & \text{if } |x| < n/2, \\ 1 & \text{otherwise,} \end{cases} \tag{2.21}$$

where $|x|$ is the Hamming weight of string x (the number of ones in it). □

2.3.2 Asymptotic bounds

Let us find the exact value of the average success probability for the optimal pure RAC suggested in Theorem 2. We will separately consider the even and odd cases.

In the odd case ($n = 2m+1$) the average success probability is given by

$$p(2m+1) = \frac{1}{(2m+1) \cdot 2^{2m+1}} \left(2 \sum_{i=m+1}^{2m+1} i \binom{2m+1}{i} \right), \tag{2.22}$$

where the factor 2 stands for either zeros or ones being the majority, and $\binom{2m+1}{i}$ stands for the number of strings where the given symbol dominates and appears exactly i times.

In the even case ($n = 2m$) there are a lot of strings with the same number of zeros and ones. These strings are bad, because with majority encoding and identity decoding it is not possible to give the correct answer for more than half of all bits. The corresponding average success probability is given by

$$p(2m) = \frac{1}{2m \cdot 2^{2m}} \left(2 \sum_{i=m+1}^{2m} i \binom{2m}{i} + m \binom{2m}{m} \right), \tag{2.23}$$

where the last term stands for the bad strings.

In Appendix A we give a combinatorial interpretation of the sums in (2.22) and (2.23). Equations (A.1) and (A.2) derived in Appendix A can be used to simplify $p(2m+1)$ and $p(2m)$, respectively. It turns out that both probabilities are equal:

$$p(2m) = p(2m+1) = \frac{1}{2} + \frac{1}{2^{2m+1}} \binom{2m}{m}. \tag{2.24}$$

These two expressions can be combined as follows:

$$p(n) = \frac{1}{2} + \frac{1}{2^n} \binom{n-1}{\lfloor \frac{n-1}{2} \rfloor}. \tag{2.25}$$

We can apply Stirling's approximation [20] $m! \approx \left(\frac{m}{e}\right)^m \sqrt{2\pi m}$ to (2.24) and obtain

$$p(2m) = p(2m+1) \approx \frac{1}{2} + \frac{1}{2\sqrt{\pi m}}. \tag{2.26}$$

If we put $m \approx \frac{n}{2}$, then (2.26) turns to

$$p(n) \approx \frac{1}{2} + \frac{1}{\sqrt{2\pi n}}. \tag{2.27}$$

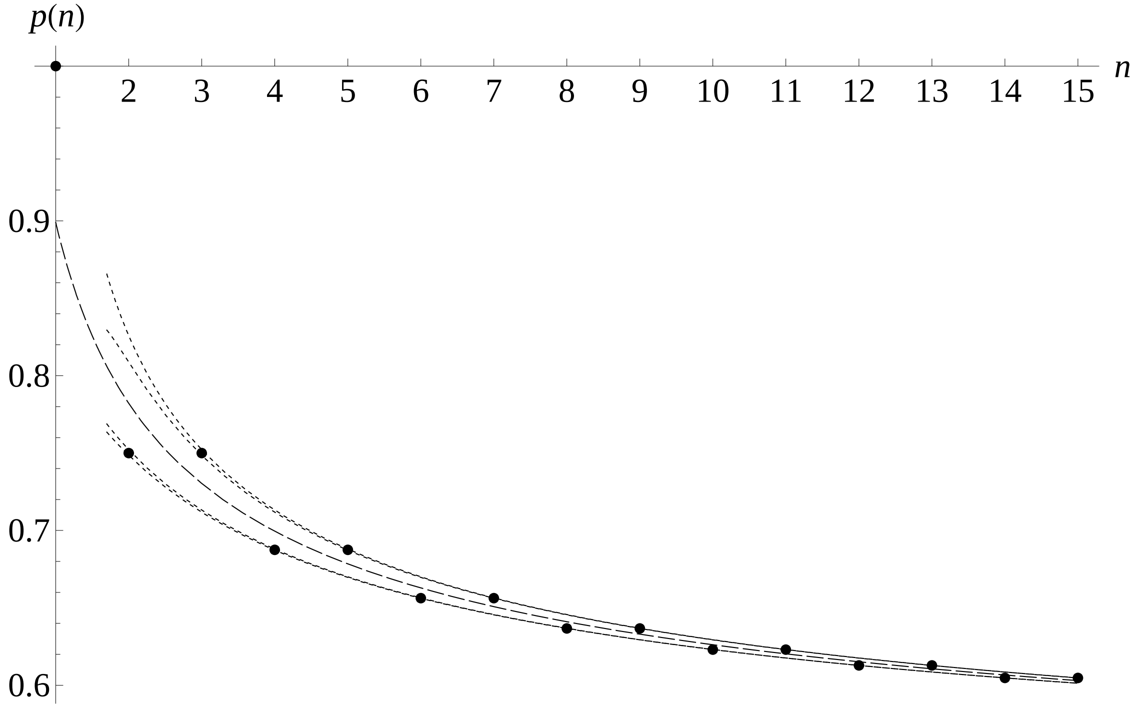

Figure 2.1: Exact probability of success $p(n)$ for optimal pure classical $n \mapsto 1$ RAC (black dots) according to (2.24) and its approximate value (dashed line) according to (2.27). Dotted lines show upper and lower bounds of $p(n)$ for odd and even n according to inequalities (2.29) and (2.30).

We see that the value of (2.27) approaches 1/2 as n increases. Thus the obtained codes are not very good for large n, since $p = 1/2$ can be obtained by guessing. We will observe a similar (but slightly better) behavior also in the quantum case. The exact probability (2.24) and its approximation (2.27) are shown in Fig. 2.1.

For odd and even cases asymptotic upper and lower bounds on $p(n)$ can be obtained using the following inequality [20]:

$$\sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{1}{12n+1}} < n! < \sqrt{2\pi n}\left(\frac{n}{e}\right)^n e^{\frac{1}{12n}}. \tag{2.28}$$

For the odd case we have

$$\frac{\exp\left(\frac{1}{12n-11} - \frac{2}{6n-6}\right)}{\sqrt{2\pi(n-1)}} < p(n) - \frac{1}{2} < \frac{\exp\left(\frac{1}{12n-12} - \frac{2}{6n-5}\right)}{\sqrt{2\pi(n-1)}}, \tag{2.29}$$

but for the even case

$$\frac{\exp\left(\frac{1}{12n} - \frac{2}{6n+1}\right)}{\sqrt{2\pi n}} < p(n) - \frac{1}{2} < \frac{\exp\left(\frac{1}{12n+1} - \frac{2}{6n}\right)}{\sqrt{2\pi n}}. \tag{2.30}$$

All four bounds are shown in Fig. 2.1.

Chapter 3

Quantum random access codes

3.1 Visualizing a qubit

When dealing with quantum random access codes (at least in the qubit case), it is a good idea to try to visualize them. We provide two ways.

3.1.1 Bloch sphere representation

A *pure qubit state* is a column vector $|\psi\rangle \in \mathbb{C}^2$. It can be expressed as a linear combination $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, where $|0\rangle = \left(\begin{smallmatrix}1\\0\end{smallmatrix}\right)$ and $|1\rangle = \left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$. The coefficients $\alpha, \beta \in \mathbb{C}$ must satisfy $|\alpha|^2 + |\beta|^2 = 1$. Since the physical state is not affected by the *phase factor* (i.e., $|\psi\rangle$ and $e^{i\phi}\,|\psi\rangle$ are the same states for any $\phi \in \mathbb{R}$), without the loss of generality one can write

$$|\psi\rangle = \begin{pmatrix} \cos\frac{\theta}{2} \\ e^{i\varphi}\sin\frac{\theta}{2} \end{pmatrix}, \tag{3.1}$$

where $0 \le \theta \le \pi$ and $0 \le \varphi < 2\pi$ (the factor 1/2 for θ in (3.1) is chosen so that these ranges resemble the ones for *spherical coordinates* in $\mathbb{R}^3$).

For almost all states $|\psi\rangle$ there is a unique way to assign the parameters θ and φ. The only exceptions are states $|0\rangle$ and $|1\rangle$, that correspond to $\theta = 0$ and $\theta = \pi$, respectively. In both cases φ does not affect the physical state. Note that the spherical coordinates with *latitude* θ and *longitude* φ have the same property, namely—the longitude is not defined at poles. This suggests that the state space of a single qubit is topologically a sphere.

Indeed, there is a one-to-one correspondence between pure qubit states and the points on a unit sphere in $\mathbb{R}^3$. This is called the *Bloch sphere representation* of a qubit state. The *Bloch vector* for state (3.1) is $\boldsymbol{r} = (x, y, z)$, where the coordinates (see Fig. 3.1) are given by

$$\begin{cases} x = \sin\theta\cos\varphi, \\ y = \sin\theta\sin\varphi, \\ z = \cos\theta. \end{cases} \tag{3.2}$$

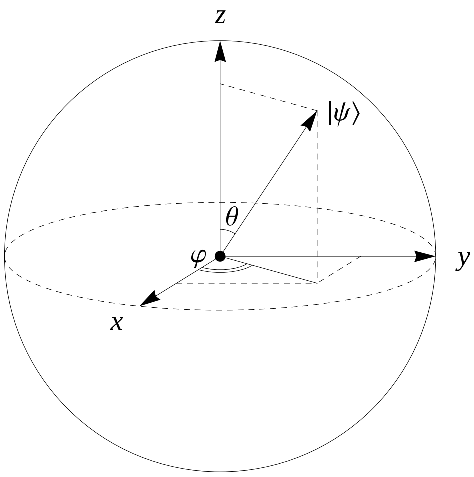

Figure 3.1: Angles θ and φ of the Bloch vector corresponding to state $|\psi\rangle$.

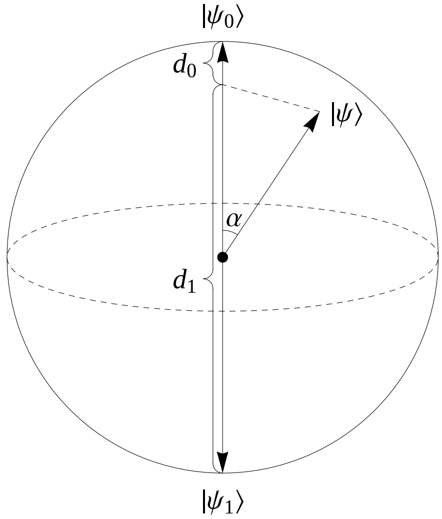

Figure 3.2: Geometric interpretation of orthogonal measurement.

Given the Bloch vector $\boldsymbol{r} = (x, y, z)$, the coefficients of the corresponding state $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ can be found as follows [19, pp. 102]:

$$\alpha = \sqrt{\frac{z+1}{2}}, \quad \beta = \frac{x+iy}{\sqrt{2(z+1)}} \tag{3.3}$$

with the convention that $(0, 0, -1)$ corresponds to $\alpha = 0$ and $\beta = 1$.

The *density matrix* of a pure state $|\psi\rangle$ is defined as $\rho = |\psi\rangle\langle\psi|$. For the state $|\psi\rangle$ in (3.1) we have

$$\rho = \frac{1}{2}\begin{pmatrix} 1+\cos\theta & e^{-i\varphi}\sin\theta \\ e^{i\varphi}\sin\theta & 1-\cos\theta \end{pmatrix} = \frac{1}{2}\left(I + x\sigma_x + y\sigma_y + z\sigma_z\right), \tag{3.4}$$

where (x, y, z) are the coordinates of the Bloch vector $\boldsymbol{r}$ given in (3.2) and

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{3.5}$$

are called *Pauli matrices*. We can write (3.4) more concisely as

$$\rho = \frac{1}{2}\left(I + \boldsymbol{r}\cdot\boldsymbol{\sigma}\right) \tag{3.6}$$

where $\boldsymbol{r} = (x, y, z)$ and $\boldsymbol{\sigma} = (\sigma_x, \sigma_y, \sigma_z)$.

If $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are the Bloch vectors of two pure states $|\psi_1\rangle$ and $|\psi_2\rangle$, then

$$|\langle\psi_1|\psi_2\rangle|^2 = \mathrm{Tr}(\rho_1\rho_2) = \frac{1}{2}(1 + \boldsymbol{r}_1\cdot\boldsymbol{r}_2). \tag{3.7}$$

This relates the inner product in $\mathbb{C}^2$ to the one in $\mathbb{R}^3$. Since $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are unit vectors, $\boldsymbol{r}_1\cdot\boldsymbol{r}_2 = \cos\alpha$, where α is the angle between $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$.

An *orthogonal measurement* M on a qubit can be specified by a set of two orthonormal states: $M = \{|\psi_0\rangle, |\psi_1\rangle\}$. Orthonormality means that $\langle\psi_i|\psi_j\rangle = \delta_{ij}$. If we measure a qubit that is in state $|\psi\rangle$ with measurement M then the *outcome* will be either 0 or 1 and the state will "collapse" to $|\psi_0\rangle$ or $|\psi_1\rangle$ with probabilities $|\langle\psi_0|\psi\rangle|^2$ and $|\langle\psi_1|\psi\rangle|^2$, respectively. Observe that for orthogonal states equation (3.7) implies $\boldsymbol{r}_1 \cdot \boldsymbol{r}_2 = -1$, therefore they correspond to antipodal points on the Bloch sphere. If we denote the angle between the Bloch vectors of $|\psi\rangle$ and $|\psi_0\rangle$ by α, then according to (3.7) the probabilities of the outcomes are

$$\begin{cases} p_0 = \dfrac{1}{2}(1 + \cos\alpha), \\ p_1 = \dfrac{1}{2}(1 - \cos\alpha). \end{cases} \tag{3.8}$$

There is a nice geometrical interpretation of these probabilities. If we project the Bloch vector corresponding to $|\psi\rangle$ on the axes spanned by the Bloch vectors of $|\psi_0\rangle$ and $|\psi_1\rangle$ (see Fig. 3.2), then $p_0 = d_1/2$ and $p_1 = d_0/2$ (note the different indices), where d_0 is the distance between the projection and $|\psi_0\rangle$, but d_1 is the distance between the projection and $|\psi_1\rangle$. Observe that vectors on the upper hemisphere have greater probability to collapse to $|\psi_0\rangle$, but on lower hemisphere, to $|\psi_1\rangle$. On the equator both probabilities are equal to $\frac{1}{2}$.

3.1.2 Unit disk representation

There is another way of visualizing a qubit. Unlike the Bloch sphere representation, this way of representing a qubit is not known to have appeared elsewhere. The idea is to use only one complex number to specify a pure qubit state $|\psi\rangle = \left(\begin{smallmatrix}\alpha \\ \beta\end{smallmatrix}\right) \in \mathbb{C}^2$. This is possible since $|\psi\rangle$ can be written in the form (3.1), which is completely determined by its second component

$$\beta = e^{i\varphi} \sin\frac{\theta}{2}. \tag{3.9}$$

The first component is just $\sqrt{1 - |\beta|^2} = \alpha$. As $|\beta| \leq 1$, the set of all possible qubit states can be identified with a unit disk in the complex plane (the polar coordinates assigned to $|\psi\rangle$ are (r, φ), where $r = \sin\frac{\theta}{2}$). The origin $\beta = 0$ corresponds to $|\psi\rangle = |0\rangle$, and all points on the unit circle $|\beta| = 1$ are identified with $|\psi\rangle = |1\rangle$, since $e^{i\varphi}|1\rangle$ corresponds to the same quantum state for all $\varphi \in \mathbb{R}$.

The relation between the unit disk representation and the Bloch sphere representation can be visualized as follows:

- the unit disk is obtained by puncturing the Bloch sphere at its South pole and flattening it,
- the Bloch sphere is obtained by gluing together the boundary of the unit disk.

It is much harder to visualize how a unitary transformation acts in the unit disk representation. Let us consider a simple example.

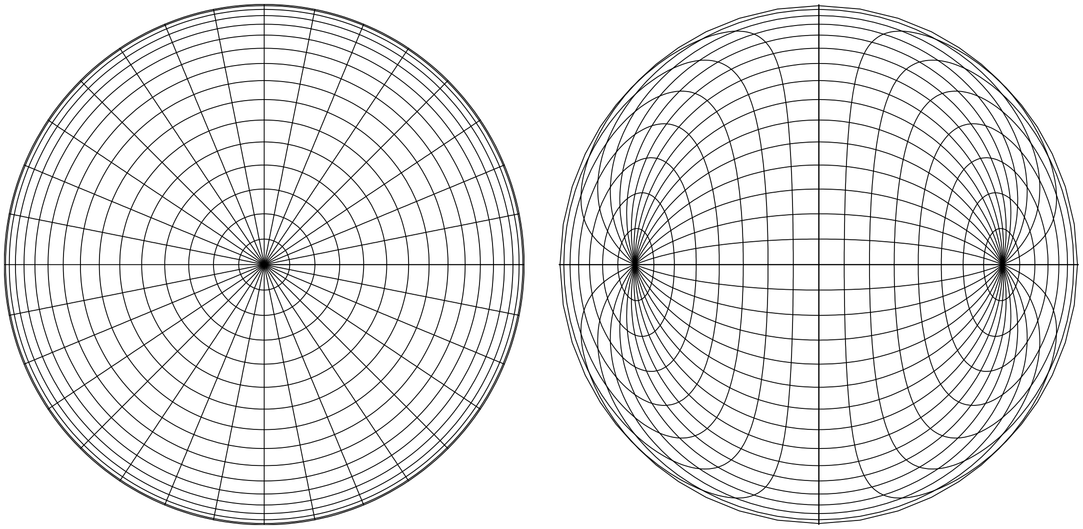

Figure 3.3: Curves of constant θ and φ before (on the left) and after the Hadamard transformation (on the right). Initially the curves of constant θ are concentric circles, but after the transformation they appear as deformed circles around both poles. The curves of constant φ transform form radial rays to "field lines" connecting both poles. The image on the left appears to have only the North pole $|0\rangle$, since the Bloch sphere is punctured at the South pole $|1\rangle$ which must be identified with the boundary of the unit disk. The "left pole" and "right pole" in the image on the right correspond to the states $|1\rangle$ and $|0\rangle$, respectively.

Example. Let us consider the action of the *Hadamard gate* $H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$ in the unit disk representation. Note that $H^2 = I$ thus H is an involution (self-inverse). It acts on the standard basis states as follows:

$$H\left|0\right\rangle = \tfrac{1}{\sqrt{2}}\left|0\right\rangle + \tfrac{1}{\sqrt{2}}\left|1\right\rangle = \left|+\right\rangle, \tag{3.10}$$

$$H\left|1\right\rangle = \tfrac{1}{\sqrt{2}}\left|0\right\rangle - \tfrac{1}{\sqrt{2}}\left|1\right\rangle = \left|-\right\rangle. \tag{3.11}$$

The way H transforms the curves of constant θ and φ is shown in Fig. 3.3. From equation (3.10) we see that the origin $\beta = 0$ corresponding to $\left|0\right\rangle$ is mapped to the "right pole" $\beta = \frac{1}{\sqrt{2}}$ corresponding to $\left|+\right\rangle$ (and vice versa). Recall that all points on the boundary of the unit disk in Fig. 3.3 (on the left) are identified with $\left|1\right\rangle$. Thus equation (3.11) tells us that the unit circle $|\beta| = 1$ is mapped to the "left pole" $\beta = -\frac{1}{\sqrt{2}}$ in Fig. 3.3 (on the right) corresponding to $\left|-\right\rangle$ (and vice versa). This means that $\left|-\right\rangle$ is mapped to the boundary of the unit disk in Fig. 3.3 (on the right).

Since we use only one complex number β to represent a quantum state, a finite set of quantum states $\{\beta_1, \beta_2, \dots, \beta_n\}$ can be represented by a polynomial

$$c\,(\beta - \beta_1)(\beta - \beta_2)\cdots(\beta - \beta_n) \tag{3.12}$$

whose roots are β_i (here $c \neq 0$ is arbitrary). We will use this representation in Sects. 3.3 and 4.1 to describe the qubit states whose Bloch vectors are the vertices of certain polyhedra. It is surprising that for those states the values of c can be chosen so that the resulting polynomials have integer coefficients.

3.2 Types of quantum encoding-decoding strategies

Let us now consider the quantum analogue of a pure strategy.

Definition. A *pure quantum* $n \mapsto 1$ *encoding-decoding strategy* is an ordered tuple $(E, M_1, \dots, M_n)$ that consists of encoding function $E : \{0,1\}^n \mapsto \mathbb{C}^2$ and n orthogonal measurements: $M_i = \{\left|\psi_0^i\right\rangle, \left|\psi_1^i\right\rangle\}$.

If Alice encodes the string x with function E, she obtains a pure qubit state $\left|\psi\right\rangle = E(x)$. When Bob receives $\left|\psi\right\rangle$ and is asked to recover the ith bit of x, he performs the measurement M_i. The probability that Bob recovers x_i correctly is equal to

$$p(x,i) = \left|\left\langle \psi_{x_i}^i \middle| \psi \right\rangle\right|^2. \tag{3.13}$$

As in the classical setting, we can allow Alice and Bob to have probabilistic quantum strategies without cooperation. Though we will not need it, mixed quantum strategies can be defined in complete analogy with mixed classical strategies.

Definition. A *mixed quantum* $n \mapsto 1$ *encoding-decoding strategy* is an ordered tuple $(P_E, P_{M_1}, \ldots, P_{M_n})$ of probability distributions. P_E is a distribution over encoding functions E and P_{M_i} are probability distributions over orthogonal measurements of qubit.

The main objects of our research are quantum strategies with cooperation, i.e., with shared randomness. They are defined in complete analogy with the classical ones.

Definition. A *quantum* $n \mapsto 1$ *encoding-decoding strategy with shared randomness* is a probability distribution over pure quantum strategies.

We would like to point out two very important things about quantum strategies with shared randomness. The first thing is that all statements about classical strategies with SR in Sect. 2.2 are valid for quantum strategies as well (the only difference is that *"pure strategy"* now means *"pure quantum strategy"* instead of *"pure classical strategy"* and *"strategy with SR"* means *"quantum strategy with SR"* instead of *"classical strategy with SR"*). The most important consequence of this observation is that Theorem 1 is valid also for quantum strategies with SR. This means that the same technique of obtaining the upper bound can be used in the quantum case, i.e., we can consider the average success probability of a pure quantum strategy instead of the worst case success probability of the quantum strategy with SR.

The second thing is that the quantum strategy with SR is the most powerful quantum encoding-decoding strategy when both kinds of classical randomness (local and shared) is allowed. However, it is not the most general strategy, since it cannot be used to simulate certain classical strategies, e.g., the ones with fixed output. However, it turns out that the ability to simulate such strategies does not give any advantage (see Sect. 3.7 and Appendix B).

3.3 Known quantum RACs

In [2, 3] it has been shown that for $2 \mapsto 1$ classical RACs in the mixed setting the decoding party cannot do better than guessing, i.e., the worst case success probability cannot exceed 1/2. However, if quantum states can be transmitted, there are pure quantum $2 \mapsto 1$ and $3 \mapsto 1$ schemes [2, 3]. This clearly indicates the advantages of quantum RACs. On the other hand, a quantum $4 \mapsto 1$ scheme cannot exist [5]. We will review these results in the next three sections.

3.3.1 The $2 \mapsto 1$ QRAC

The $2 \mapsto 1$ QRAC is described in [2, 3, 5]. The main idea is to use two mutually orthogonal pairs of antipodal Bloch vectors for measurement bases. For example, let M_1 and M_2 be the measurements along the x and y axes, respectively. The corresponding Bloch vectors are $\boldsymbol{v}_1 = (\pm 1, 0, 0)$ and $\boldsymbol{v}_2 = (0, \pm 1, 0)$. The measurement

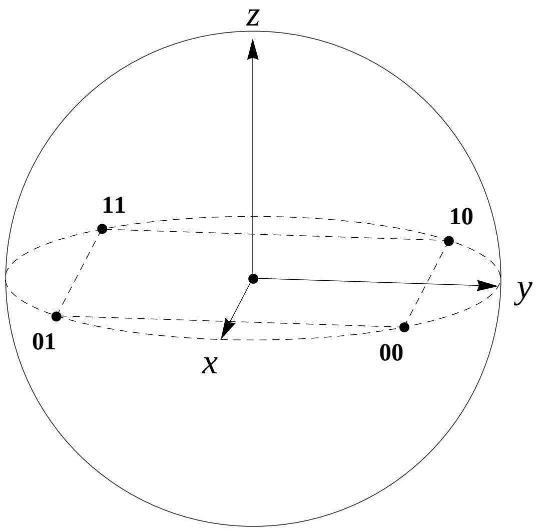

Figure 3.4: Bloch sphere representation of encoding for 2 ↦ 1 quantum random access code.

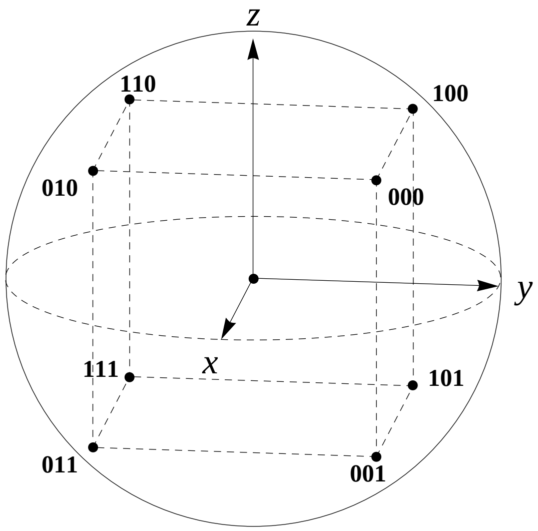

Figure 3.5: Bloch sphere representation of encoding for 3 ↦ 1 quantum random access code.

bases are

$$M_1 = \left\{ \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right\}, \tag{3.14}$$

$$M_2 = \left\{ \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ i \end{pmatrix}, \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -i \end{pmatrix} \right\}. \tag{3.15}$$

The planes orthogonal to the x and y axes cut the Bloch sphere into four parts. Note that in each part only one definite string can be encoded (otherwise the worst case success probability will be less than $\frac{1}{2}$). According to (3.8), all encoding points must be as far from both planes as possible in order to maximize the worst case success probability (recall the geometrical interpretation of the measurement shown in Fig. 3.2). In our case the best encoding states are the vertices of a square $\frac{1}{\sqrt{2}}(\pm 1, \pm 1, 0)$ inscribed in the unit circle on the xy plane (see Fig. 3.4). Given a string $x = x_1 x_2$, the Bloch vector of the encoding state can be found as follows:

$$\boldsymbol{r}(x) = \frac{1}{\sqrt{2}} \begin{pmatrix} (-1)^{x_1} \\ (-1)^{x_2} \\ 0 \end{pmatrix}. \tag{3.16}$$

The corresponding encoding function is

$$E(x_1, x_2) = \frac{1}{\sqrt{2}} |0\rangle + \frac{(-1)^{x_1} + i(-1)^{x_2}}{2} |1\rangle. \tag{3.17}$$

The success probability is the same for all input strings and all bits to be recovered:

$$p = \frac{1}{2} \left(1 + \cos \frac{\pi}{4} \right) = \frac{1}{2} + \frac{1}{2\sqrt{2}} \approx 0.853553. \tag{3.18}$$

3.3.2 The $3 \mapsto 1$ QRAC

It is not hard to generalize the $2 \mapsto 1$ QRAC to a $3 \mapsto 1$ code—just take three mutually orthogonal pairs of antipodal Bloch vectors, i.e., the vertices of an *octahedron* [5, 8]. The third pair is $\boldsymbol{v}_3 = (0, 0, \pm 1)$ and the corresponding measurement basis is

$$M_3 = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}. \tag{3.19}$$

In this case we have three orthogonal planes that cut the sphere into eight parts and only one string can be encoded into each part. In this case the optimal encoding states correspond to the vertices of a cube $\frac{1}{\sqrt{3}}(\pm 1, \pm 1, \pm 1)$ inscribed in the Bloch sphere (see Fig. 3.5). The Bloch vector of the encoding state of string $x = x_1 x_2 x_3$ is

$$\boldsymbol{r}(x) = \frac{1}{\sqrt{3}} \begin{pmatrix} (-1)^{x_1} \\ (-1)^{x_2} \\ (-1)^{x_3} \end{pmatrix}. \tag{3.20}$$

The corresponding encoding function is $E(x_1, x_2, x_3) = \alpha \left|0\right\rangle + \beta \left|1\right\rangle$ with coefficients α and β explicitly given by

$$\begin{cases} \alpha = \sqrt{\dfrac{1}{2} + \dfrac{(-1)^{x_3}}{2\sqrt{3}}}, \\ \beta = \dfrac{(-1)^{x_1} + i(-1)^{x_2}}{\sqrt{6 + 2\sqrt{3}(-1)^{x_3}}}. \end{cases} \tag{3.21}$$

In fact, the coefficients β are exactly the eight roots of the polynomial[1]

$$36\beta^8 + 24\beta^4 + 1 \tag{3.22}$$

This code also has the same success probability in all cases:

$$p = \frac{1}{2} + \frac{1}{2\sqrt{3}} \approx 0.788675. \tag{3.23}$$

3.3.3 Impossibility of the $4 \mapsto 1$ QRAC

Hayashi et al. [5] have shown that $2 \mapsto 1$ and $3 \mapsto 1$ codes discussed above cannot be generalized for 4 (and hence more) encoded bits. The reason is simple—it is not possible to cut the Bloch sphere into 16 parts with 4 great circles (see the proof below). Thus the number of strings will exceed the number of parts, hence at least two strings must be encoded in the same part. This makes the worst case success probability drop below $\frac{1}{2}$.

Let us consider how many parts can be obtained by cutting a sphere with 4 great circles. Without loss of generality we can assume that the first great circle

[1] The unit disk representation of a quantum state and the representation of a finite set of quantum states using a polynomial was discussed in Sect. 3.1.2.

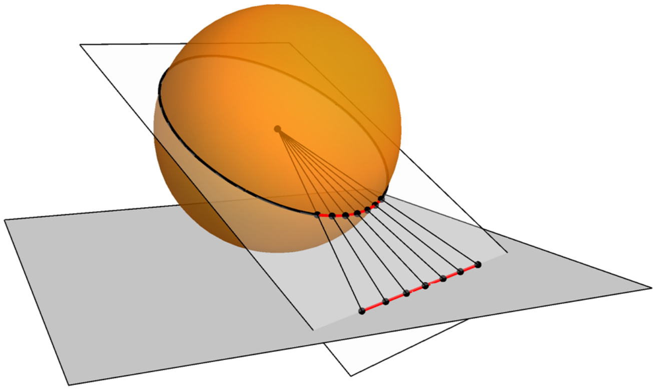

Figure 3.6: Gnomonic projection transforms great circles to lines and vice versa.

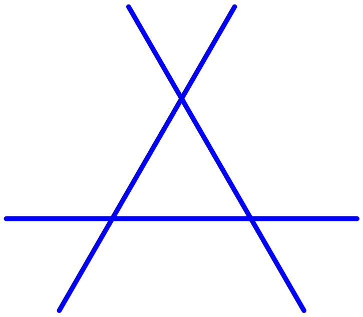

Figure 3.7: Cutting the plane with 3 lines into 7 parts.

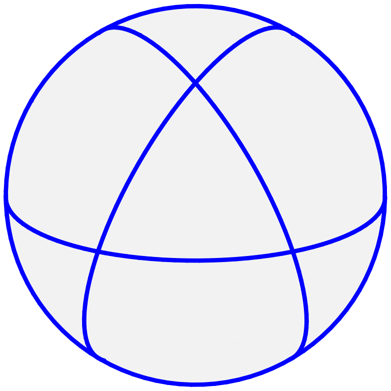

Figure 3.8: Cutting the sphere with 4 great circles into 14 parts (seven diametrically opposite parts are equal).

coincides with the equator. We use the *gnomonic projection* (from the center of the sphere) to project the remaining three circles to a plane tangent to the South pole. Note that great circles are transformed into lines and vice versa (see Fig. 3.6), thus we will obtain three lines. Also note that each region in the plane corresponds to two (diametrically opposite) regions on the sphere. It is simple to verify that three lines cannot cut the plane into more than 7 parts (see Fig. 3.7). Thus the sphere cannot be cut into more than 14 parts with four great circles.[2] An example achieving the upper bound is shown in see Fig. 3.8 (see also Figs. 4.15 and 4.16). Using essentially the same argument for generalized Bloch vectors Hayashi et al. [5] show that $2^{2m} \overset{p}{\mapsto} m$ QRACs with $p > 1/2$ do not exist for all $m \geq 1$. The generalized Bloch vector will be briefly introduced in Sect. 5.3.

[2] In general, if we have n great circles on the sphere, the maximal number of parts we can obtain is twice what we can obtain by cutting the plane with $n-1$ lines. If each line we draw intersects all previous lines and no three lines intersect at the same point, the sphere is cut into $n(n-1)+2$ parts after the inverse gnomonic projection.

3.4 Optimal encoding for given decoding strategy

We just reviewed the known results on pure $n \mapsto 1$ quantum random access codes. From now on we will consider QRACs with shared randomness. In this section we will show how to find the optimal encoding strategy for a given decoding strategy. More precisely, we will show that the measurement directions of a QRAC with SR determine the corresponding optimal encoding states in a simple way.

An orthogonal measurement for the ith bit is specified by antipodal points on the Bloch sphere: $M_i = \{\boldsymbol{v}_i, -\boldsymbol{v}_i\}$. Let $\boldsymbol{r}_x$ be the Bloch vector that corresponds to the quantum state in which string $x \in \{0,1\}^n$ is encoded. According to equations in (3.8) the success probability for input (x, i) is

$$p(x,i) = \frac{1}{2}\big(1 + (-1)^{x_i} \boldsymbol{v}_i \cdot \boldsymbol{r}_x\big) \tag{3.24}$$

and the average success probability is given by

$$\begin{aligned} p &= \frac{1}{2^n \cdot n} \sum_{x \in \{0,1\}^n} \sum_{i=1}^{n} \frac{1}{2}\big(1 + (-1)^{x_i} \boldsymbol{v}_i \cdot \boldsymbol{r}_x\big) \\ &= \frac{1}{2}\left(1 + \frac{1}{2^n \cdot n} \underbrace{\sum_{x \in \{0,1\}^n} \boldsymbol{r}_x \cdot \sum_{i=1}^{n} (-1)^{x_i} \boldsymbol{v}_i}_{S_{\boldsymbol{v},\boldsymbol{r}}}\right). \end{aligned} \tag{3.25}$$

In order to maximize the probability p, we only need to maximize $S_{\boldsymbol{v},\boldsymbol{r}}$ in equation (3.25) over all possible measurements $\boldsymbol{v}_i$ and encodings $\boldsymbol{r}_x$ (in total $n + 2^n$ unit vectors in $\mathbb{R}^3$). We will denote the maximum of $S_{\boldsymbol{v},\boldsymbol{r}}$ by $S(n)$:

$$S(n) \;=\; \max_{\{\boldsymbol{v}_i\},\{\boldsymbol{r}_x\}} S_{\boldsymbol{v},\boldsymbol{r}} \;=\; \max_{\{\boldsymbol{v}_i\}} \sum_{x \in \{0,1\}^n} \max_{\boldsymbol{r}_x} \; \boldsymbol{r}_x \cdot \sum_{i=1}^{n} (-1)^{x_i} \boldsymbol{v}_i. \tag{3.26}$$

If we define

$$\boldsymbol{v}_x = \sum_{i=1}^{n} (-1)^{x_i} \boldsymbol{v}_i, \tag{3.27}$$

then it is obvious that the scalar product $\boldsymbol{r}_x \cdot \boldsymbol{v}_x$ in (3.26) will be maximized when $\boldsymbol{r}_x$ is chosen along the same direction as $\boldsymbol{v}_x$, i.e. $\boldsymbol{r}_x = \boldsymbol{v}_x / \left\|\boldsymbol{v}_x\right\|$ when $\left\|\boldsymbol{v}_x\right\| \neq 0$. In this case we have $\boldsymbol{r}_x \cdot \boldsymbol{v}_x = \left\|\boldsymbol{v}_x\right\|$ and

$$S(n) = \max_{\{\boldsymbol{v}_i\}} \sum_{x \in \{0,1\}^n} \left\| \sum_{i=1}^{n} (-1)^{x_i} \boldsymbol{v}_i \right\|. \tag{3.28}$$

Therefore we only need to maximize over all possible measurements succinctly represented by n unit vectors $\boldsymbol{v}_i \in \mathbb{R}^3$, because the optimal encoding is already determined by measurements (see Sect. 4.1 for some numerical results obtained in this way). When the value of $S(n)$ is found, then according to (3.25) the corresponding probability is

$$p(n) = \frac{1}{2}\left(1 + \frac{S(n)}{2^n \cdot n}\right). \tag{3.29}$$

We can observe a connection between quantum and classical RACs with SR. Assume that Marge and Homer[3] have to implement $n \mapsto 1$ QRAC with SR and are deciding what strategies to use—Homer is responsible for choosing the measurements, but Marge has to choose how to encode the input string. Once they have decided, they have to follow the agreement and cannot cheat. Unfortunately, Homer is foolish and he proposes to measure all bits in the same basis. Luckily Marge is clever enough to choose the optimal encoding for Homer's measurements. According to the discussion above, she has to use the majority encoding function. Thus the obtained QRAC is as good as an optimal classical RAC discussed in Sect. 2.3.1, Theorem 2.

It looks plausible that using the same measurement for all bits is the worst decoding strategy. However, we have not proved this, so we leave it as a conjecture:

Conjecture. *For any choice of measurements there is an encoding such that the resulting $n \mapsto 1$ quantum RAC with SR is at least as good as the optimal $n \mapsto 1$ classical one.*

3.5 Relation to a random walk in $\mathbb{R}^3$

QRACs with shared randomness are related to random walks in $\mathbb{R}^3$. This relation can be seen by suitably interpreting equations (3.28) and (3.29). Let us consider an $n \mapsto 1$ QRAC with SR whose measurement directions are given by unit vectors $\{\boldsymbol{v}_i\}$ and let us assume that the corresponding optimal encoding for these measurements is used as described in the previous section. Then we can write the success probability $p(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n)$ of this QRAC in the following suggestive form:

$$p(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n) = \frac{1}{2}\Big(1 + \frac{1}{n}\, d(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n)\Big), \tag{3.30}$$

where

$$d(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n) = \frac{1}{2^n} \sum_{a \in \{1,-1\}^n} \left\| \sum_{i=1}^{n} a_i \boldsymbol{v}_i \right\| \tag{3.31}$$

is the average distance traveled by a random walk whose ith step is $\boldsymbol{v}_i$ or $-\boldsymbol{v}_i$, each with probability 1/2. For example, $\boldsymbol{v}_1 = \boldsymbol{v}_2 = \cdots = \boldsymbol{v}_n$ corresponds to a random walk on a line and $d(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n)$ is the average distance traveled after n steps of this walk. Recall from the previous section that this choice of $\{\boldsymbol{v}_i\}$ corresponds to the optimal classical RAC and we conjecture that this is the worst possible choice. Similarly, if we choose roughly one third of vectors $\{\boldsymbol{v}_i\}$ along each coordinate

[3] In this scenario it is more convenient to replace Alice and Bob with Marge and Homer from *The Simpsons.*

axis, we obtain a random walk in a cubic lattice and $d(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n)$ is the average distance traveled when roughly $n/3$ steps are performed along each coordinate axis (see Sect. 3.8.2).

In Sects. 3.8 we will use this relation between random access codes and random walks to prove a lower bound for the success probability of $n \mapsto 1$ QRACs with SR.

3.6 Upper bound

In this section we will derive an upper bound for $S(n)$. For this purpose we rewrite the equation (3.28) in the following form:

$$S(n) = \max_{\{\boldsymbol{v}_i\}} S_{\boldsymbol{v}} \tag{3.32}$$

where

$$S_{\boldsymbol{v}} = \sum_{a \in \{1,-1\}^n} \left\| \sum_{i=1}^{n} a_i \boldsymbol{v}_i \right\| \tag{3.33}$$

(for convenience we take the sum over the set $\{1,-1\}^n$ instead of $\{0,1\}^n$).

Lemma 5. *For any unit vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ we have*

$$\sum_{a_1,\ldots,a_n \in \{1,-1\}} \|a_1 \boldsymbol{v}_1 + \cdots + a_n \boldsymbol{v}_n\|^2 = n \cdot 2^n. \tag{3.34}$$

Proof. For $n = 1$ we have

$$\sum_{a_1 \in \{1,-1\}} \|a_1 \boldsymbol{v}_1\|^2 = \|\boldsymbol{v}_1\|^2 + \|-\boldsymbol{v}_1\|^2 = 2. \tag{3.35}$$

Let us assume that equation (3.34) holds for $n = k$. Then for $n = k + 1$ we have

$$\sum_{a_1,\ldots,a_k,a_{k+1} \in \{1,-1\}} \|a_1 \boldsymbol{v}_1 + \cdots + a_k \boldsymbol{v}_k + a_{k+1} \boldsymbol{v}_{k+1}\|^2 . \tag{3.36}$$

If we write out the sum over a_{k+1} explicitly, we obtain

$$\sum_{a_1,\ldots,a_k \in \{1,-1\}} \left(\|a_1 \boldsymbol{v}_1 + \cdots + a_k \boldsymbol{v}_k + \boldsymbol{v}_{k+1}\|^2 + \|a_1 \boldsymbol{v}_1 + \cdots + a_k \boldsymbol{v}_k - \boldsymbol{v}_{k+1}\|^2 \right) . \tag{3.37}$$

We can use the parallelogram identity

$$\|\boldsymbol{u}_1 + \boldsymbol{u}_2\|^2 + \|\boldsymbol{u}_1 - \boldsymbol{u}_2\|^2 = 2 \left(\|\boldsymbol{u}_1\|^2 + \|\boldsymbol{u}_2\|^2 \right) , \tag{3.38}$$

which holds for any two vectors $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, to simplify the sum as follows:

$$\sum_{a_1,\ldots,a_k \in \{1,-1\}} 2 \left(\|a_1 \boldsymbol{v}_1 + \cdots + a_k \boldsymbol{v}_k\|^2 + \|\boldsymbol{v}_{k+1}\|^2 \right) . \tag{3.39}$$

We know that $\boldsymbol{v}_{k+1}$ is a unit vector and we have assumed that (3.34) holds for $n = k$; therefore (3.39) simplifies to $2 \left(k \cdot 2^k + 2^k \right) = (k + 1) \cdot 2^{k+1}$. □

We will use the previous lemma to obtain an upper bound for $S_{\boldsymbol{v}}^2$ defined in (3.33). According to (3.32) this will give us an upper bound for $S(n)$ as well.

Lemma 6. *For any set of unit vectors $\{\boldsymbol{v}_i\}_{i=1}^n$, the inequality $S_{\boldsymbol{v}} \leq \sqrt{n} \cdot 2^n$ holds.*

Proof. We can interpret the first sum in equation (3.33) as an inner product with $(1, \dots, 1) \in \mathbb{R}^{2^n}$. Then the Cauchy-Schwarz inequality $\boldsymbol{x} \cdot \boldsymbol{y} \leq \|\boldsymbol{x}\| \, \|\boldsymbol{y}\|$ says that

$$S_{\boldsymbol{v}} \leq \sqrt{2^n} \sqrt{\sum_{a \in \{1,-1\}^n} \left\| \sum_{i=1}^n a_i \boldsymbol{v}_i \right\|^2} = \sqrt{2^n} \sqrt{n \cdot 2^n} = \sqrt{n} \cdot 2^n, \tag{3.40}$$

where Lemma 5 was used to obtain the first equality. □

Theorem 3. *For any $n \overset{p}{\mapsto} 1$ QRAC with shared randomness, $p \leq \frac{1}{2} + \frac{1}{2\sqrt{n}}$.*

Proof. From Lemma 6 we have $S_{\boldsymbol{v}} \leq \sqrt{n} \cdot 2^n$. From equation (3.32) we see that the same upper bound applies to $S(n)$. Putting this into (3.29) we get

$$p \leq \frac{1}{2} + \frac{1}{2\sqrt{n}}.$$

□

In particular, this means that the known $2 \mapsto 1$ and $3 \mapsto 1$ QRACs discussed in Sect. 3.3 cannot be improved even if shared randomness is allowed.

The intuition behind this upper bound is as follows. If instead of $\mathbb{R}^3$ the Bloch vector of a qubit state would be in $\mathbb{R}^n$, we could choose all n measurements to be mutually orthogonal. For example, we could choose the vectors forming measurement bases to be the vertices of the *cross polytope*, i.e., all permutations of $(\pm 1, 0, \dots, 0)$. The optimal encoding corresponding to this choice are the vertices of the *hypercube*, i.e., points $(\pm 1, \pm 1, \dots, \pm 1)$, thus all terms in equation (3.33) are equal to $\sqrt{n}$ and sum to $2^n\sqrt{n}$, so the probability (3.29) is $\frac{1}{2}(1 + \frac{2^n\sqrt{n}}{2^n n}) = \frac{1}{2} + \frac{1}{2\sqrt{n}}$. Since we have only three dimensions, the actual probability should not be larger.

3.7 General upper bound

Let us prove an analogue of Theorem 3 for a more general model, because quantum mechanics allows us to consider more general quantum states and measurements. Namely, Alice can encode her message into a *mixed state* instead of a pure state and Bob can use a POVM measurement instead of an orthogonal measurement to recover information. A mixed state is just a probability distribution over pure states, so it does not provide a more general encoding model. In contrast, a POVM measurement provides a more general decoding model. In fact, there is another reason to extend the model.

Example. It is not possible to construct a pure QRAC (as defined in Sect. 3.2) that simulates the following pure classical $2 \mapsto 1$ RAC:

- *encoding:* encode the first bit,

- *decoding:* if the first bit is requested, output the received bit; if the second one is requested—output 0 no matter what is received.

To recover the first bit with certainty, Alice and Bob have to agree on two antipodal points on the Bloch sphere, where the information is encoded. Unfortunately the second bit will cause a problem—it is not possible to choose an orthogonal measurement of a qubit in an unknown state, so that the result is always the same.

This example suggests that the model of pure quantum encoding-decoding strategies introduced in Sect. 3.2 should be extended in one way or the other. It is obvious that a *constant decoding function* (0 or 1) can be implemented using a single-outcome POVM measurement. However, it turns out that in the qubit case a two-outcome POVM measurement can be replaced by a probability distribution over orthogonal measurements and constant decoding functions (see appendix B). This means that both extensions are equivalent. For simplicity we choose to extend the model by allowing constant decoding functions, thus Bob can either perform an orthogonal measurement or use a constant decoding function. The goal of this section is to show that constant decoding functions do not give any advantage.

Definition. An *enhanced orthogonal measurement* is either an orthogonal measurement or one that always gives the same answer.

Definition. An *enhanced pure quantum $n \mapsto 1$ encoding-decoding strategy* is an ordered tuple $(E, M_1, \ldots, M_n)$ consisting of encoding function $E : \{0,1\}^n \mapsto \mathbb{C}^2$ and n decoding functions M_i that are enhanced orthogonal measurements.

Definition. An *enhanced quantum encoding-decoding strategy with SR* is a probability distribution over enhanced pure quantum strategies.

Now it is straightforward to construct a pure quantum RAC for the previous example. In fact, now any classical RAC (either pure, mixed or with SR) can be simulated by the corresponding type of a quantum RAC.

There is no need to further extend the model of enhanced QRACs with SR by adding other types of classical randomness. For example, a probabilistic combination of POVMs does not provide a more general measurement, because it can be simulated by a probabilistic combination of enhanced orthogonal measurements. The same holds for probabilistic post-processing of the measurement results (which can be simulated by a probabilistic combination of enhanced orthogonal measurements as shown in Appendix B). Therefore enhanced QRACs with SR constitute the most general model when any kind of classical randomness is allowed.

One might suspect that the upper bound obtained in Theorem 3 does not hold for this model, but this is not the case.

Theorem 4. *For any $n \overset{p}{\mapsto} 1$ enhanced QRAC with SR, $p \leq \frac{1}{2} + \frac{1}{2\sqrt{n}}$.*

Proof. According to Yao's principle and Theorem 1, we can consider the average success probability of pure enhanced QRACs instead. It suffices to rule out the constant decoding functions. More precisely, we have to show that QRACs having a constant decoding function for some bit give a smaller upper bound than those without it. In fact, we are proving a quantum analogue of Lemma 3 from Sect. 2.3.1.

We will use induction on n. The case $n = 1$ is trivial—a pure enhanced QRAC with a constant decoding function has average success probability $\frac{1}{2} < 1$. Let us assume that for some $n = k - 1 \geq 1$ the constant decoding functions do not give any benefit. We now prove that the same holds for $n = k$. Let us assume that the constant decoding function 0 is used for the kth bit. The average case success probability is

$$p(k) = \frac{1}{2^k \cdot k} \sum_{x \in \{0,1\}^k} \left(\sum_{i=1}^{k-1} p(x,i) + \delta_{0,x_k} \right), \tag{3.41}$$

where $p(x, i)$ is the success probability (3.13) for the input (x, i) where $i \leq k - 1$ and δ_{0,x_k} is the probability that the decoding function 0 gives a correct answer for the kth bit. The last bit can be ignored during the encoding and decoding of other bits:

$$p(k) = \left(\frac{1}{2^k \cdot k} \sum_{x \in \{0,1\}^{k-1}} 2 \sum_{i=1}^{k-1} p(x,i) \right) + \frac{1}{2k} \tag{3.42}$$

$$= \frac{k-1}{k} \left(\frac{1}{2^{k-1} \cdot (k-1)} \sum_{x \in \{0,1\}^{k-1}} \sum_{i=1}^{k-1} p(x,i) \right) + \frac{1}{2k}. \tag{3.43}$$

Note that the bracketed expression in (3.43) is the success probability $p(k-1)$ of a shorter QRAC. Therefore

$$p(k) = \frac{k-1}{k} \cdot p(k-1) + \frac{1}{2k}. \tag{3.44}$$

Now we can apply the inductive hypothesis:

$$p(k) \leq \frac{k-1}{k} \left(\frac{1}{2} + \frac{1}{2\sqrt{k-1}} \right) + \frac{1}{2k} = \frac{1}{2} + \frac{\sqrt{k-1}}{2k} < \frac{1}{2} + \frac{1}{2\sqrt{k}}, \tag{3.45}$$

completing the proof. Thus the upper bound obtained in Theorem 3 holds for the general model as well. □

Observe again that for $n = 2$ and $n = 3$ this upper bound matches equations (3.18) and (3.23), respectively. This means that the known $2 \mapsto 1$ and $3 \mapsto 1$ quantum random access codes with pure encoding-decoding strategies (see Sects. 3.3.1 and 3.3.2, respectively) are optimal even among enhanced strategies with SR. For $n = 4$ we get $p \leq \frac{3}{4}$.

A similar upper bound was recently obtained by Ben-Aroya et al. [16] for $n \overset{p}{\mapsto} m$ QRACs, where k bits must be recovered. They allow randomized strategies without shared randomness. In particular, they show that for any $\eta > 2 \ln 2$ there exists a constant C_η such that for $n \gg k$

$$p \leq C_\eta \left(\frac{1}{2} + \frac{1}{2}\sqrt{\frac{\eta m}{n}} \right)^k. \tag{3.46}$$

It might be possible to generalize our upper bound (3.45) to obtain something similar to (3.46).

3.8 Lower bounds

In the next two sections we will describe two constructions of $n \mapsto 1$ QRAC with SR for all $n \geq 1$. These constructions provide a lower bound on the success probability. They use random and orthogonal measurements, respectively. In the first case it is hard to compute the exact success probability even for small values of n, but we will obtain an asymptotic expression. However, in the second case we do not know the asymptotic success probability, but can easily compute the exact success probability for small n.

3.8.1 Lower bound by random measurements

We now turn to lower bound for p. A lower bound for QRACs with shared randomness can be obtained by randomized encoding. Alice and Bob can use the shared random string to agree on some random orthogonal measurement for each bit. Each of these measurement bases can be specified by antipodal points on the Bloch sphere (see Sect. 3.1.1). These points can be sampled by using some sphere point picking method [21], near uniformly given enough shared randomness. The chosen measurements determine the optimal encoding scheme (see Sect. 3.4) which is known to both sides.

The expected success probability of randomized $n \mapsto 1$ QRAC similarly to (3.30) is given by

$$\mathbb{E}(p) = \frac{1}{2}\left(1 + \frac{1}{n} \mathop{\mathbb{E}}_{\{\boldsymbol{v}_i\}} d(\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n)\right) \tag{3.47}$$

where according to equation (3.31)

$$\begin{aligned} \mathop{\mathbb{E}}_{\{\boldsymbol{v}_i\}} d(\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n) &= \mathop{\mathbb{E}}_{\{\boldsymbol{v}_i\}} \left(\frac{1}{2^n} \sum_{a \in \{1,-1\}^n} \left\| \sum_{i=1}^{n} a_i \boldsymbol{v}_i \right\| \right) && (3.48) \\ &= \frac{1}{2^n} \sum_{a \in \{1,-1\}^n} \mathop{\mathbb{E}}_{\{\boldsymbol{v}_i\}} \left\| \sum_{i=1}^{n} a_i \boldsymbol{v}_i \right\|. && (3.49) \end{aligned}$$

Each $a \in \{1, -1\}^n$ influences the direction of some vectors $\boldsymbol{v}_i$, but the resulting set $\{a_i \boldsymbol{v}_i\}$ is still uniformly distributed. Therefore the expected value in equation (3.49) does not depend on a and we have

$$\mathop{\mathbb{E}}_{\{\boldsymbol{v}_i\}} d(\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n) = \mathop{\mathbb{E}}_{\{\boldsymbol{v}_i\}} \left\| \sum_{i=1}^{n} \boldsymbol{v}_i \right\|. \tag{3.50}$$

This expression has a very nice geometrical interpretation—it is the average distance traveled by a particle that performs n steps of unit length each in a random direction. This distance can be found by evaluating the following integral:

$$\frac{1}{(4\pi)^n}\int_{\theta_1=0}^{\pi}\int_{\varphi_1=0}^{2\pi}\cdots\int_{\theta_n=0}^{\pi}\int_{\varphi_n=0}^{2\pi}\left\|\sum_{i=1}^{n}\begin{pmatrix}\sin\theta_i\cos\varphi_i\\ \sin\theta_i\sin\varphi_i\\ \cos\theta_i\end{pmatrix}\right\|\prod_{i=1}^{n}\sin\theta_i\,d\theta_i\,d\varphi_i. \tag{3.51}$$

Unfortunately it is very hard to evaluate it even numerically, since the integrand is highly oscillatory. An alternative approach is to directly simulate a random walk by sampling points uniformly from the sphere [21]. For small values of n the success probability (3.47) averaged over 10^6 simulations is given in Table 3.1. Luckily, we have the following asymptotic result:

Theorem 5 (Chandrasekhar [22, pp. 14], Hughes [23, pp. 91])**.** *The probability density to arrive at point $\boldsymbol{R}$ after performing $n \gg 1$ steps of random walk is*

$$W(\boldsymbol{R}) \approx \left(\frac{3}{2\pi n}\right)^{3/2}\exp\left(-\frac{3\left\|\boldsymbol{R}\right\|^2}{2n}\right). \tag{3.52}$$

Theorem 6. *For every $n \gg 1$ there exists an $n \overset{p}{\mapsto} 1$ QRAC with expected success probability $p \approx \frac{1}{2} + \sqrt{\frac{2}{3\pi n}}$.*

Proof. Because of the spherical symmetry of the probability density in formula (3.52), the average distance traveled after $n \gg 1$ steps of random walk is given by

$$\mathop{\mathbb{E}}_{\{\boldsymbol{v}_i\}}\left\|\sum_{i=1}^{n}\boldsymbol{v}_i\right\| \approx \int_0^\infty R\cdot W(R)\cdot 4\pi R^2\,dR = 2\sqrt{\frac{2n}{3\pi}}. \tag{3.53}$$

From (3.50) and (3.47) we obtain

$$\mathbb{E}(p) \approx \frac{1}{2} + \sqrt{\frac{2}{3\pi n}}, \tag{3.54}$$

which gives the desired lower bound. □

Formally this lower bound holds only for large n. However, if one estimates the actual value of (3.53) by random sampling one can see that the asymptotic expression (3.54) is indeed smaller than the actual value (see Table. 3.1).

3.8.2 Lower bound by orthogonal measurements

According to the upper bound obtained in Sect. 3.6 the known $2 \mapsto 1$ and $3 \mapsto 1$ QRACs (see Sect. 3.3) are optimal. This suggests that orthogonal measurements can be used to construct good codes. Unfortunately this idea cannot be directly applied when $n > 3$, since in $\mathbb{R}^3$ there are only three mutually orthogonal directions. However, if we choose roughly one third of all measurements along each coordinate axis, we will get quite a lot of mutually orthogonal measurement pairs.

Let $\boldsymbol{v}_1 = (1,0,0)$, $\boldsymbol{v}_2 = (0,1,0)$, $\boldsymbol{v}_3 = (0,0,1)$, and $\forall i : \boldsymbol{v}_{i+3} \equiv \boldsymbol{v}_i$. According to equation (3.30) in Sect. 3.5 the success probability of this $n \mapsto 1$ QRAC with SR is related to the average distance (3.31) traveled by a random walk. For our choice of measurement directions $\boldsymbol{v}_i$ the random walk takes place in a cubic lattice and consists of roughly $n/3$ steps along each coordinate axis. Thus we can simplify the equation (3.31) for the average distance traveled to avoid having an exponential number of terms in it:

$$d(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n) = \frac{1}{2^n} \sum_{i=0}^{x} \sum_{j=0}^{y} \sum_{k=0}^{z} \binom{x}{i} \binom{y}{j} \binom{z}{k} \sqrt{(x-2i)^2 + (y-2j)^2 + (z-2k)^2}, \quad (3.55)$$

where $x + y + z = n$ and each of x, y, z is roughly $n/3$. The corresponding success probability can be obtained by plugging this expression in equation (3.30).

This lower bound is better than the one obtained in the previous section using random measurements and it also requires less shared randomness. The difference of both lower bounds is shown in Fig. 3.9. The periodic pattern of length 6 in this picture can be explained as follows. When n is a multiple of 3, the same number of steps of a random walk is performed along each coordinate axis (this explains the factor 3). To explain the factor 2, let us consider a random walk on a line, i.e., one of the three coordinate axis. The distinction between odd an even number of steps of such a walk is that the probability distribution after an even number of steps is peaked at the origin, but this peak has no contribution whatsoever to the average distance traveled. This intuition suggests that it should be especially hard to beat this lower bound when n is of the form $6k + 3$.

	Random measurements		Orthogonal measurements	
n	Asymptotic	Sampling	Numerical	Exact
2	0.825735	0.8333	0.853553	$\frac{1}{2} + \frac{1}{2\sqrt{2}}$
3	0.765962	0.7708	0.788675	$\frac{1}{2} + \frac{1}{2\sqrt{3}}$
4	0.730329	0.7333	0.741481	$\frac{1}{2} + \frac{1+\sqrt{3}}{8\sqrt{2}}$
5	0.706013	0.7082	0.711803	$\frac{1}{2} + \frac{2+\sqrt{5}}{20}$
6	0.688063	0.6897	0.686973	$\frac{1}{2} + \frac{1+\sqrt{3}+\sqrt{6}}{16\sqrt{3}}$
7	0.674113	0.6754	0.677458	$\frac{1}{2} + \frac{15+6\sqrt{5}+2\sqrt{13}+\sqrt{17}}{224}$
8	0.662868	0.6638	0.666270	$\frac{1}{2} + \frac{12+9\sqrt{3}+6\sqrt{5}+6\sqrt{7}+\sqrt{11}}{256\sqrt{2}}$
9	0.653553	0.6544	0.656893	$\frac{1}{2} + \frac{10\sqrt{3}+9\sqrt{11}+3\sqrt{19}}{384}$

Table 3.1: Comparison of $n \mapsto 1$ QRACs with SR that use random and orthogonal measurements, respectively. For the first code we give the success probability according to the asymptotic expression (3.54) and a numerical value obtained by 10^6 random samples. For the second code we give both the numerical and the exact value of the success probability according to equation (3.55).

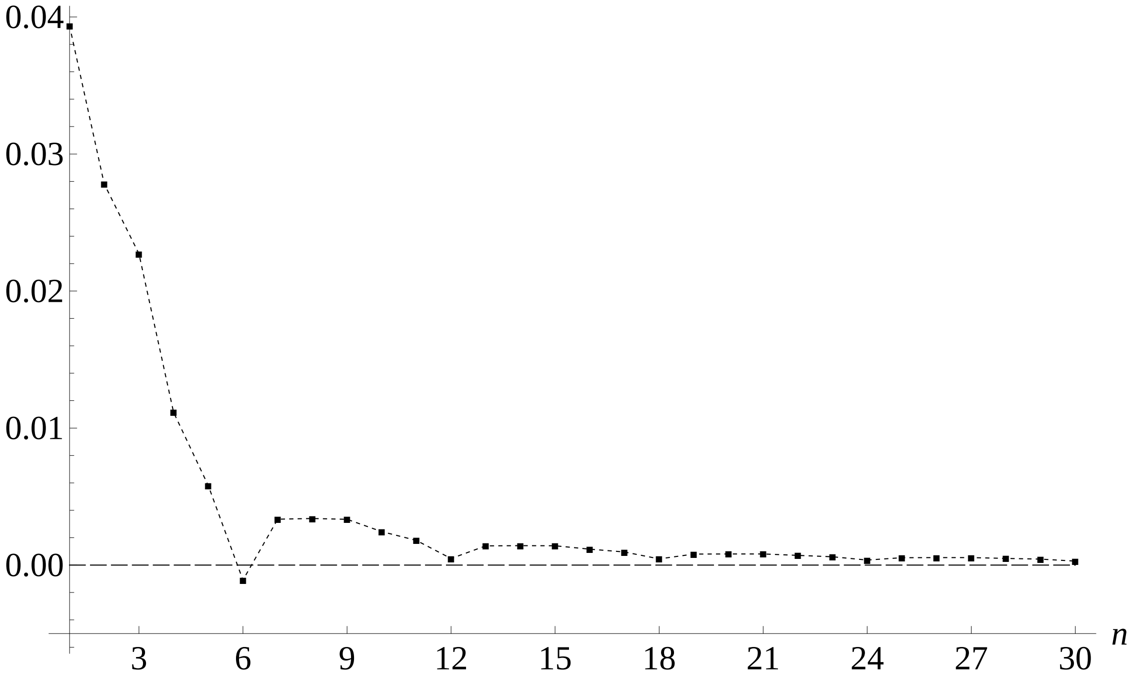

Figure 3.9: The *difference* of both lower bounds for QRACs with SR. Black squares correspond to the bound obtained using measurements along coordinate axes and the horizontal line corresponds to the asymptotic bound (3.54) using random measurements (see Sects. 3.8.1 and 3.8.2, respectively). The first bound is better, except for $n = 6$ (notice a periodic pattern of length 6).

Chapter 4

Constructions of QRACs with SR

It is plausible that one can do better than the lower bound obtained above, which used random measurements. In this section we will consider several constructions of quantum random access codes with shared randomness for some particular values of n. First, in Sect. 4.1 we will describe numerically obtained QRACs. Then, in Sect. 4.2 we will construct new QRACs with high degree of symmetry. In Sect. 4.3 we will compare both kinds of codes and draw some conclusions.

4.1 Numerical results

n	Section	Probability
2	4.1.1	0.853553
3	4.1.1	0.788675
4	4.1.2	0.741481
5	4.1.3	0.713578
6	4.1.4	0.694046
7		0.678638
8		0.666633
9	4.1.5	0.656893
10		0.648200
11		0.641051
12		0.634871

Table 4.1: The success probabilities of numerical $n \mapsto 1$ QRACs.

In this section we will discuss some particular $n \mapsto 1$ QRACs with shared randomness for several small values of n. These codes were obtained using numerical optimization. The optimization must be performed only over all possible measurements, because in Sect. 3.4 we showed that the choice of measurements determines the optimal encoding in a simple way. Each measurement is specified by a unit vector $\boldsymbol{v}_i \in \mathbb{R}^3$. For $n \mapsto 1$ QRAC there are n such vectors and one needs two angles to specify each of them. Without loss of generality we can assume that $\boldsymbol{v}_1 = (0, 0, 1)$

due to the rotational symmetry of the Bloch sphere. Thus only $2(n-1)$ real parameters are required to specify all $\boldsymbol{v}_i$ and therefore an $n \mapsto 1$ QRAC. To find the best configuration of measurements $\boldsymbol{v}_i$, one needs to maximize $S_{\boldsymbol{v}}$ given by (3.33). According to (3.29) the success probability of the corresponding QRAC is given by

$$p_{\boldsymbol{v}} = \frac{1}{2}\left(1 + \frac{S_{\boldsymbol{v}}}{2^n \cdot n}\right). \tag{4.1}$$

This is not a convex optimization problem, since the feasible set (given by $\|\boldsymbol{v}_i\| = 1$ for all $1 \leq i \leq n$) is not convex. Note that it is not convex even if we relax equalities $\|\boldsymbol{v}_i\| = 1$ to inequalities $\|\boldsymbol{v}_i\| \leq 1$. We used the *Mathematica*'s general-purpose built-in function `NMaximize` to solve this problem.

Once the measurements $\boldsymbol{v}_i$ are found, one can easily obtain the Bloch vector $\boldsymbol{r}_x$ of the qubit state that must be used to optimally encode the string x. We showed (see Sect. 3.4) that $\boldsymbol{r}_x$ is a unit vector in direction $\boldsymbol{v}_x$, where $\boldsymbol{v}_x$ is given by (3.27). For almost all QRACs that we have found using numerical optimization, the points $\boldsymbol{r}_x$ form a symmetric pattern on the surface of the Bloch ball. Thus we were able to guess the exact values of $\boldsymbol{r}_x$ and $\boldsymbol{v}_i$. However, as in any numerical optimization, optimality of the resulting codes is not guaranteed.

In order to make the resulting codes more understandable, we depict them in three dimensions using the following conventions:

- each *red point* encodes the string indicated,
- each *blue point* defines the axis of the measurement when the indicated bit is to be output, and
- for each measurement there is a corresponding (unlabeled) *blue great circle* containing states yielding 0 and 1 equiprobably.

More precisely, the blue point with label i defines the basis vector $|\psi_0^i\rangle$ corresponding to the outcome 0 of the ith measurement (see Sect. 3.2). Note that the blue circles and blue points come in pairs—the vector $|\psi_0^i\rangle$ defined by the blue point is orthogonal to the corresponding circle. As a cautionary note, occasionally, the blue point for one measurement falls on the great circle of a different measurement (for example, blue points 1 and 2 in Fig. 4.2 lie on one another's corresponding circles). If there are too many red points, we omit the string labels for clarity.

Usually the codes have some symmetry; for example, the encoding points may be the vertices of a polyhedron. In such cases we show the corresponding polyhedron instead of the Bloch sphere. We do not discuss $7 \mapsto 1$ and $8 \mapsto 1$ QRACs since the best numerical results have almost no discernible symmetry. We also do not discuss the numerical results for $n \geq 10$ (see Table 4.1 for success probabilities). The numerically obtained $10 \mapsto 1$ code is symmetric and resembles $6 \mapsto 1$ code discussed in Sect. 4.1.4, but the $11 \mapsto 1$ and $12 \mapsto 1$ codes again have almost no discernible symmetry. Success probabilities of all numerical $n \mapsto 1$ QRACs with SR are summarized in Table 4.1 and Fig. 4.1.

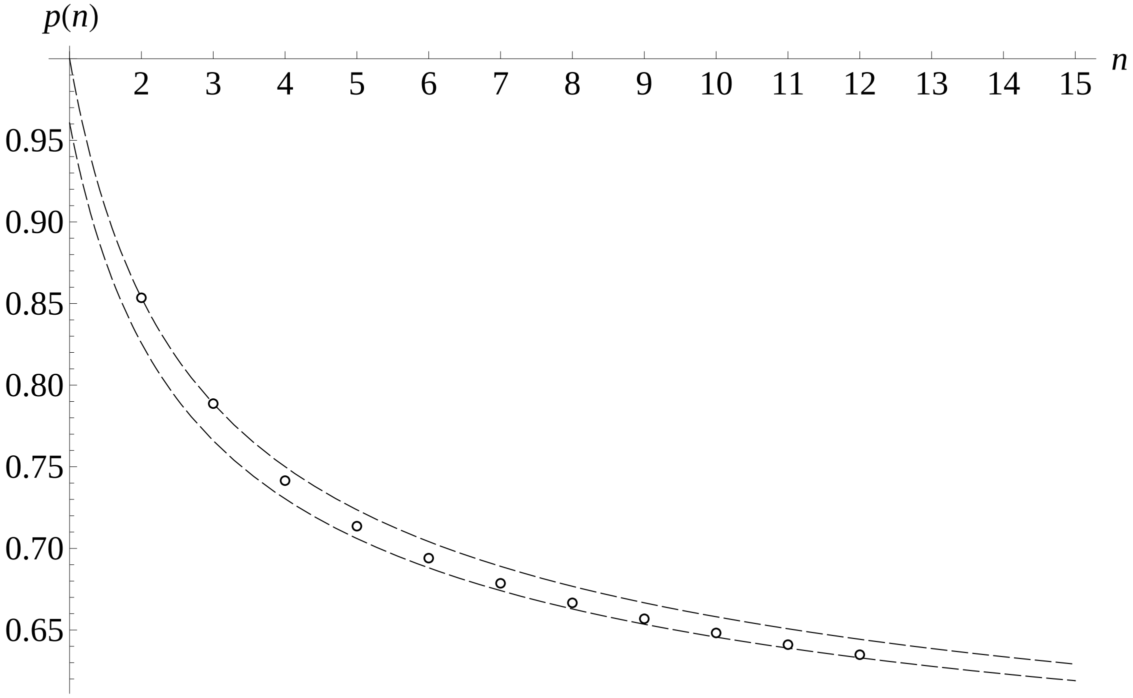

Figure 4.1: Success probabilities $p(n)$ of numerical $n \mapsto 1$ QRACs with SR from Table 4.1. The upper bound $\frac{1}{2} + \frac{1}{2\sqrt{n}}$ and the lower bound $\frac{1}{2} + \sqrt{\frac{2}{3\pi n}}$ are indicated by dashed lines (see Sects. 3.7 and 3.8.1, respectively).

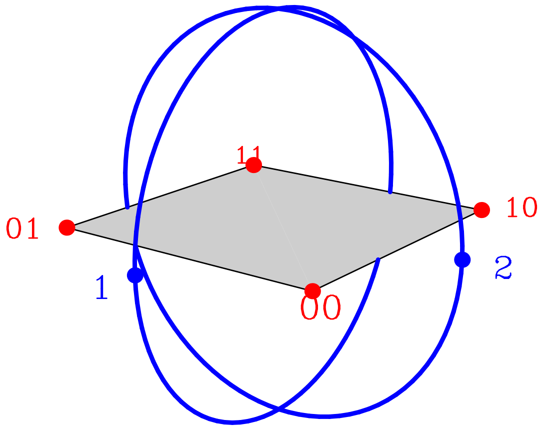

Figure 4.2: The $2 \mapsto 1$ QRAC with SR.[a]

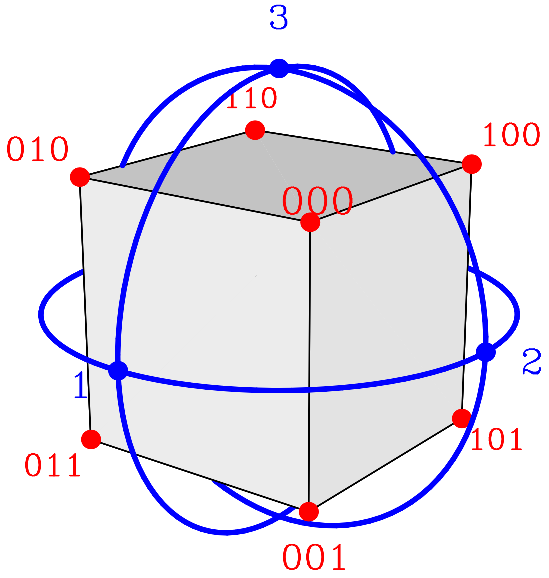

Figure 4.3: The $3 \mapsto 1$ QRAC with SR.

4.1.1 The $2 \mapsto 1$ and $3 \mapsto 1$ QRACs with SR

We used numerical optimization as described above to find $2 \mapsto 1$ and $3 \mapsto 1$ QRACs with shared randomness and obtained the optimal codes discussed in Sects. 3.3.1 and 3.3.2.

The codes are shown in Fig. 4.2 and 4.3, respectively. In the first case the encoding points are the vertices of a square and the success probability is

$$p = \frac{1}{2} + \frac{1}{2\sqrt{2}} \approx 0.853553. \tag{4.2}$$

In the second case they are the vertices of a cube. The success probability is

$$p = \frac{1}{2} + \frac{1}{2\sqrt{3}} \approx 0.788675. \tag{4.3}$$

4.1.2 The $4 \mapsto 1$ QRAC with SR

In Sect. 3.3.3 we discussed the impossibility of a $4 \mapsto 1$ QRAC when Alice and Bob are not allowed to cooperate. However, a $4 \mapsto 1$ QRAC can be obtained if they have shared randomness. The particular $4 \mapsto 1$ QRAC with SR discussed here was found by a numerical optimization. It is a hybrid of the $2 \mapsto 1$ and $3 \mapsto 1$ codes discussed in Sects. 3.3.1 and 3.3.2, respectively.

The measurements are performed in the bases (M_1, M_2, M_3, M_3), where M_1, M_2, and M_3 are the same as in the $3 \mapsto 1$ case (note that the last two bits are measured in the same basis, namely M_3). These bases are given by (3.14), (3.15),

[a] For those who are using a black-and-white printout: this is how **red** and **blue** looks like.

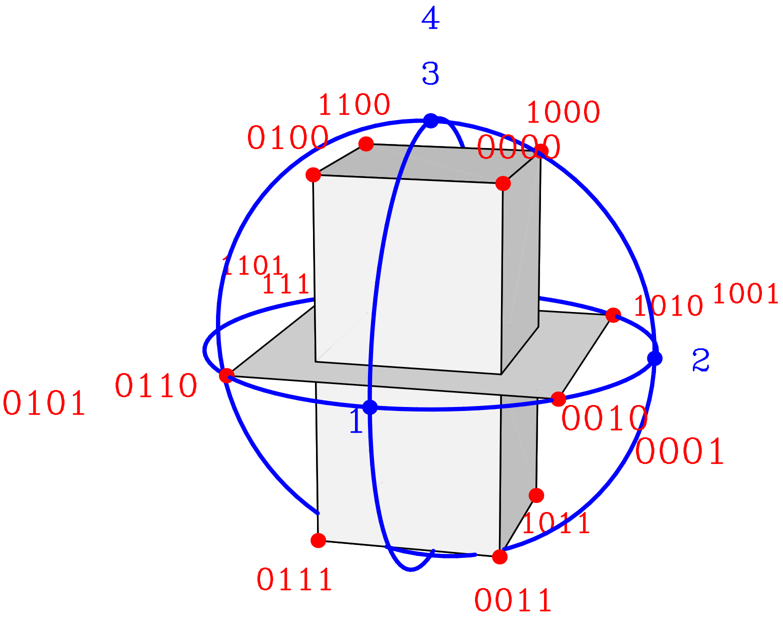

Figure 4.4: The 4 ↦ 1 QRAC with SR.

and (3.19), respectively. The points that correspond to an optimal encoding for these bases are the vertices of a regular square $\frac{1}{\sqrt{2}}(\pm 1, \pm 1, 0)$ in the xy plane and a cube $\frac{1}{\sqrt{6}}(\pm 1, \pm 1, \pm 2)$ that is stretched in the z direction (see the Bloch sphere in Fig. 4.4). The Bloch vector for the string $x = x_1 x_2 x_3 x_4$ is explicitly given by

$$\boldsymbol{r}(x) = \frac{1}{\sqrt{6}} \begin{pmatrix} (-1)^{x_1} \bigl(1 - (1 - \sqrt{3})\,|x_3 - x_4|\bigr) \\ (-1)^{x_2} \bigl(1 - (1 - \sqrt{3})\,|x_3 - x_4|\bigr) \\ (-1)^{x_3} + (-1)^{x_4} \end{pmatrix}. \tag{4.4}$$

The encoding function can be described as follows:

- if $x_3 = x_4$, use the usual 3 ↦ 1 QRAC with an emphasis on x_3 to encode the string $x_1 x_2 x_3$,
- if $x_3 \neq x_4$—encode only $x_1 x_2$ using the usual 2 ↦ 1 QRAC.

In the 3 ↦ 1 scheme the probability to recover x_3 must be increased by stretching the cube along the z axis, because x_3 equals x_4 and therefore it is of greater value than x_1 or x_2.

This 4 ↦ 1 QRAC can also be seen as a combination of two 3 ↦ 1 QRACs: the string $x_1 x_2 x_3$ is encoded into the vertices of a smaller cube inscribed in a half of the Bloch ball (the vertices that lie within the sphere are projected to its surface). The last bit x_4 indicates in which half the smaller cube lies (the upper and lower hemispheres correspond to $x_4 = 0$ and 1, respectively).

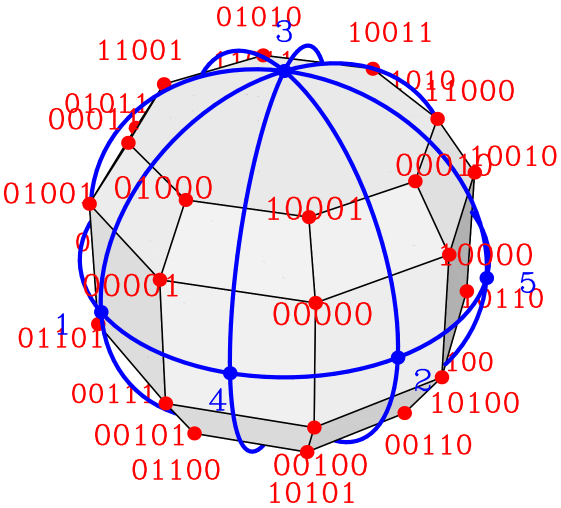

Figure 4.5: The 5 ↦ 1 QRAC with SR.

The qubit state is explicitly given by $E(x_1, x_2, x_3, x_4) = \alpha\,|0\rangle + \beta\,|1\rangle$, where

$$\begin{cases} \alpha = \sqrt{\dfrac{1}{2} + \dfrac{(-1)^{x_3} + (-1)^{x_4}}{2\sqrt{6}}}, \\ \beta = \dfrac{(-1)^{x_1} + i(-1)^{x_2}}{\sqrt{4\bigl(3 - 2\,|x_3 - x_4|\bigr) + 2\sqrt{6}\bigl((-1)^{x_3} + (-1)^{x_4}\bigr)}}. \end{cases} \tag{4.5}$$

The 16 values for β are exactly the sixteen roots of the polynomial (recall Sect. 3.1.2)

$$2304\beta^{16} + 3072\beta^{12} + 1120\beta^{8} + 128\beta^{4} + 1. \tag{4.6}$$

If a shared random string is not available, the worst case success probability of this QRAC is $\frac{1}{2}$. However, if shared randomness is available, input randomization (as in Lemma 1) can be used and we will get the same success probability for all inputs, namely

$$p = \frac{1}{2} + \frac{1 + \sqrt{3}}{8\sqrt{2}} \approx 0.741481. \tag{4.7}$$

We do not know if this 4 ↦ 1 QRAC with SR is optimal.

4.1.3 The 5 ↦ 1 QRAC with SR

To obtain a 5 ↦ 1 QRAC, we take the bases M_1, M_2, and M_3, given by (3.14), (3.15), and (3.19), respectively, and also

$$M_4 = \left\{ \frac{1}{2}\begin{pmatrix} \sqrt{2} \\ i + 1 \end{pmatrix}, \frac{1}{2}\begin{pmatrix} -\sqrt{2} \\ i + 1 \end{pmatrix} \right\}, \tag{4.8}$$

$$M_5 = \left\{ \frac{1}{2}\begin{pmatrix} \sqrt{2} \\ i - 1 \end{pmatrix}, \frac{1}{2}\begin{pmatrix} -\sqrt{2} \\ i - 1 \end{pmatrix} \right\}. \tag{4.9}$$

The Bloch vectors $\boldsymbol{v}_3 = (0, 0, \pm 1)$ for the basis M_3 are along the z axis, but the Bloch vectors of the other four bases form a regular octagon in the xy plane (shown in Fig. 4.5): $\boldsymbol{v}_1 = (\pm 1, 0, 0)$, $\boldsymbol{v}_2 = (0, \pm 1, 0)$, $\boldsymbol{v}_4 = \pm\frac{1}{\sqrt{2}}(1, 1, 0)$, $\boldsymbol{v}_5 = \pm\frac{1}{\sqrt{2}}(-1, 1, 0)$. The Bloch vector encoding the string $x = x_1x_2x_3x_4x_5$ is

$$\boldsymbol{r}(x) = \frac{1}{\sqrt{10 + s(x)4\sqrt{2}}} \begin{pmatrix} \sqrt{2}(-1)^{x_1} + (-1)^{x_4} - (-1)^{x_5} \\ \sqrt{2}(-1)^{x_2} + (-1)^{x_4} + (-1)^{x_5} \\ \sqrt{2}(-1)^{x_3} \end{pmatrix}, \tag{4.10}$$

where $s(x) \in \{-1, 1\}$ and is given by

$$s(x) = \frac{(-1)^{x_1} + (-1)^{x_2}}{2}(-1)^{x_4} - \frac{(-1)^{x_1} - (-1)^{x_2}}{2}(-1)^{x_5}. \tag{4.11}$$

The great circles with equiprobable outcomes of the measurements partition the Bloch sphere into 16 equal spherical triangles. There are two strings encoded into each triangle. The idea for how to locate the correct point for the given string x is as follows. Observe that the strings with $x_3 = 0$ and $x_3 = 1$ are encoded into the upper and lower hemisphere, respectively (this means that for all strings the probability that the measurement M_3 gives the correct value of x_3 is greater than $\frac{1}{2}$). Next observe that half of all strings have $s(x) = 1$, but the other half have $s(x) = -1$ (in fact, the two strings in the same triangle have distinct values of s).

Let us first consider the case $s(x) = 1$. We call such string *compatible* with the measurements, because it can be encoded in such a way that every measurement gives the correct value of the corresponding bit with probability greater than $\frac{1}{2}$. For the ith bit of x we can define the "preferable region" on the Bloch sphere as the hemisphere where M_i recovers x_i with probability greater than $\frac{1}{2}$. The intersection of these five regions is one sixteenth of the Bloch sphere—the triangle where x must be encoded. The point with the smallest absolute value of the z coordinate in this triangle must be chosen (it has smaller probability than other points in the triangle to recover x_3 correctly, but the probabilities for the other four bits are larger).

If $s(x) = -1$, the string x is *incompatible* with the measurements, because the intersection of the "preferable regions" is empty. Thus, no matter where the string is encoded, at least one bit will differ from the most probable outcome of the corresponding measurement. We can take this into account and modify the definition of the "preferable region" for the ith bit ($i \neq 3$). It is a union of eight triangles: four triangles where the most probable outcome of M_i equals x_i, and four triangles where it does not equal x_i (in either case the triangles with maximal probability of correct outcome of M_i must be taken). For example, the "preferable regions" for x_2 are shown in Fig. 4.6. The regions for x_3 remain the same as in the previous case. The intersection of all five regions for the given string x is the triangle where the string must be encoded. The point in the triangle with the largest absolute value of the z coordinate must be chosen. As a result, three of the measurements will give the correct value of the corresponding bit of the string x with probability greater than $\frac{1}{2}$.

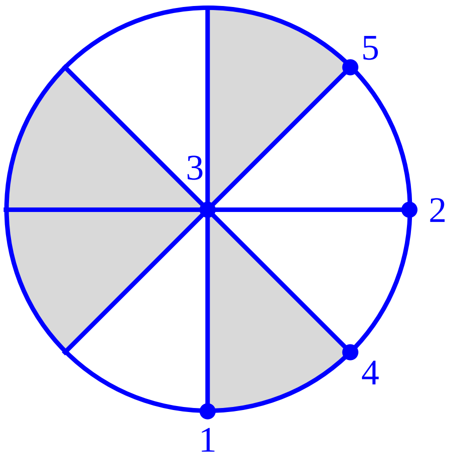

Figure 4.6: The "preferable regions" of the measurement M_2 (only the upper hemisphere is shown, the other half is symmetric). For each of the measurements the direction of the Bloch vector $|\psi_0\rangle$ is indicated by the corresponding number. The white triangles correspond to $x_2 = 0$, but the gray ones to $x_2 = 1$.

The corresponding qubit state is given by $E(x_1, x_2, x_3, x_4, x_5) = \alpha\,|0\rangle + \beta\,|1\rangle$ with coefficients α and β defined as follows:

$$\begin{cases} \alpha = \sqrt{\dfrac{1}{2} + \dfrac{(-1)^{x_3}}{2\sqrt{5 + s(x)2\sqrt{2}}}}, \\ \beta = \dfrac{(-1)^{x_1} + i(-1)^{x_2} + \frac{i+1}{\sqrt{2}}(-1)^{x_4} + \frac{i-1}{\sqrt{2}}(-1)^{x_5}}{\sqrt{10 + s(x)4\sqrt{2} + 2(-1)^{x_3}\sqrt{5 + s(x)2\sqrt{2}}}}. \end{cases} \tag{4.12}$$

The coefficients β are the roots of the polynomial

$$1336336\beta^{32} + 961792\beta^{24} + 151432\beta^{16} + 1600\beta^{8} + 1. \tag{4.13}$$

Again, using input randomization we obtain the same success probability for any input, namely

$$p = \frac{1}{2} + \frac{1}{20}\sqrt{2(5 + \sqrt{17})} \approx 0.713578. \tag{4.14}$$

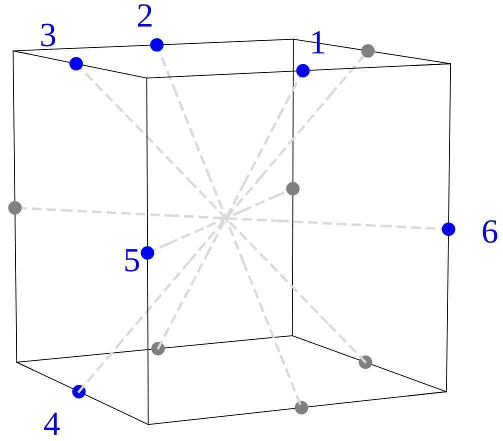

Figure 4.7: The measurements for the $6 \mapsto 1$ QRAC shown on the right.

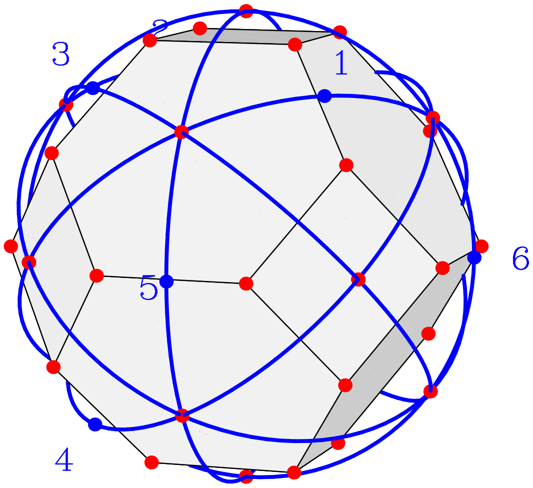

Figure 4.8: The $6 \mapsto 1$ QRAC with SR.

4.1.4 The $6 \mapsto 1$ QRAC with SR

The Bloch vectors corresponding to the 6 measurements are as follows:

$$\begin{aligned}
\boldsymbol{v}_1 &= \pm(0,+1,+1)/\sqrt{2},\\
\boldsymbol{v}_2 &= \pm(0,-1,+1)/\sqrt{2},\\
\boldsymbol{v}_3 &= \pm(+1,0,+1)/\sqrt{2},\\
\boldsymbol{v}_4 &= \pm(+1,0,-1)/\sqrt{2},\\
\boldsymbol{v}_5 &= \pm(+1,+1,0)/\sqrt{2},\\
\boldsymbol{v}_6 &= \pm(-1,+1,0)/\sqrt{2}.
\end{aligned} \tag{4.15}$$

They correspond to the 12 vertices of the *cuboctahedron* (or the midpoints of the 12 edges of the cube) and are shown in Fig. 4.7. The great circles orthogonal to these vectors form the projection of the edges of a normalized[1] *tetrakis hexahedron* and partition the Bloch sphere into 24 parts (see Fig. 4.8). Each of these parts contains one vertex of a *truncated octahedron*—the dual of tetrakis hexahedron. It is inscribed in the Bloch sphere shown in Fig. 4.8.

[1] The vertices of the *tetrakis hexahedron* are not all at the same distance from the origin (the ones forming an octahedron are $2/\sqrt{3}$ times closer than those forming a cube). So the polyhedron has to be *normalized* to fit inside the Bloch sphere (the vectors pointing to the vertices have to be rescaled to have a unit norm).

The measurement bases corresponding to $\boldsymbol{v}_i$ can be found using (3.3):

$$\begin{aligned}
M_1 &= \left\{ \frac{1}{2} \begin{pmatrix} \sqrt{2+\sqrt{2}} \\ i\sqrt{2-\sqrt{2}} \end{pmatrix}, \frac{1}{2} \begin{pmatrix} \sqrt{2-\sqrt{2}} \\ -i\sqrt{2+\sqrt{2}} \end{pmatrix} \right\}, \\
M_2 &= \left\{ \frac{1}{2} \begin{pmatrix} \sqrt{2+\sqrt{2}} \\ -i\sqrt{2-\sqrt{2}} \end{pmatrix}, \frac{1}{2} \begin{pmatrix} \sqrt{2-\sqrt{2}} \\ i\sqrt{2+\sqrt{2}} \end{pmatrix} \right\}, \\
M_3 &= \left\{ \frac{1}{2} \begin{pmatrix} \sqrt{2+\sqrt{2}} \\ \sqrt{2-\sqrt{2}} \end{pmatrix}, \frac{1}{2} \begin{pmatrix} \sqrt{2-\sqrt{2}} \\ -\sqrt{2+\sqrt{2}} \end{pmatrix} \right\}, \\
M_4 &= \left\{ \frac{1}{2} \begin{pmatrix} \sqrt{2-\sqrt{2}} \\ \sqrt{2+\sqrt{2}} \end{pmatrix}, \frac{1}{2} \begin{pmatrix} \sqrt{2+\sqrt{2}} \\ -\sqrt{2-\sqrt{2}} \end{pmatrix} \right\}, \\
M_5 &= \left\{ \frac{1}{2} \begin{pmatrix} \sqrt{2} \\ i+1 \end{pmatrix}, \frac{1}{2} \begin{pmatrix} \sqrt{2} \\ -i-1 \end{pmatrix} \right\}, \\
M_6 &= \left\{ \frac{1}{2} \begin{pmatrix} \sqrt{2} \\ i-1 \end{pmatrix}, \frac{1}{2} \begin{pmatrix} \sqrt{2} \\ -i+1 \end{pmatrix} \right\}.
\end{aligned} \tag{4.16}$$

Note that M_5 and M_6 are the same as (4.8) and (4.9) for the $5 \mapsto 1$ QRAC described in the previous section. Another way to describe these 6 bases is to consider the β coefficients for the 12 vectors that form them. It turns out that these coefficients are exactly the roots of the polynomial

$$256\beta^{12} - 128\beta^{8} - 44\beta^{4} + 1. \tag{4.17}$$

Let us consider how to determine the point where a given string should be encoded. According to (3.27) we have to find the sum of the vectors $\boldsymbol{v}_i$ defined in (4.15), each taken with either a plus or a minus sign. These vectors correspond to six pairs of opposite edges of a cube and the signs determine which edge from each pair we are taking (see Fig. 4.7). There are only three distinct ways of doing this (see Fig. 4.9). Regardless of which way it is, for each of the chosen edges there is exactly one other that shares a common face and is parallel to it. Thus we can partition the chosen edges into three pairs (in Fig. 4.9 such pairs are joined with a thick blue line). The sum of the vectors $\boldsymbol{v}_i$ for edges in a pair is always parallel to one of the axes and its direction is indicated with an arrow in Fig. 4.9. From these arrows one can see where the encoding point should lie.

Now let us classify all $2^6 = 64$ strings of length 6 into 3 types according to the location of the encoding point on the Bloch sphere. Each type of string is encoded into a vertex of a specific polyhedron (see Fig. 4.10). These polyhedra are the *cube*, the *truncated octahedron*, and the *octahedron* and the number of strings of each type are 16, 24, and 24, respectively. Let us consider them case by case:

- The *cube* has 8 vertices:

$$\frac{1}{\sqrt{3}}(\pm 1, \pm 1, \pm 1) \tag{4.18}$$

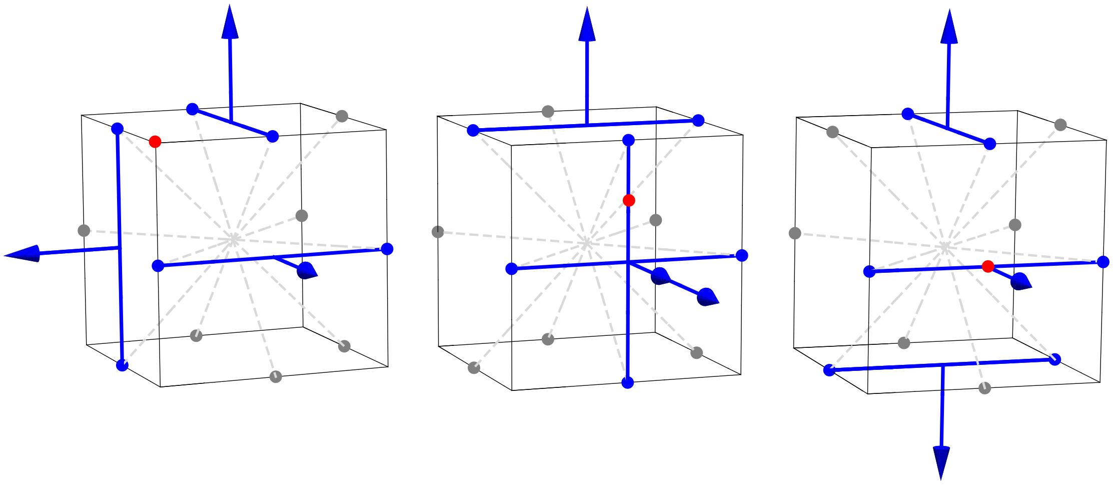

Figure 4.9: Three distinct ways of choosing one edge from each pair of opposite edges of a cube. The chosen edges are marked with blue points. Points lying on opposite edges of the same face are connected and the direction of the sum of the corresponding vectors is indicated with an arrow. The corresponding encoding point is shown in red. The red points obtained from all possible choices of the same kind are the vertices of a cube, a truncated octahedron, and an octahedron, respectively (see Fig. 4.10).

and there are 2 strings encoded into each vertex. These 16 strings are exactly those $x_1 x_2 \dots x_6 \in \{0,1\}^6$ that satisfy

$$|x_1 - x_2| + |x_3 - x_4| + |x_5 - x_6| \in \{0, 3\}\,. \tag{4.19}$$

This condition ensures that the three arrows in Fig. 4.9 are orthogonal.

- The *truncated octahedron* has 24 vertices. Their coordinates are obtained by all permutations of the components of

$$\frac{1}{\sqrt{5}}(0, \pm 1, \pm 2). \tag{4.20}$$

Truncated octahedron	Octahedron
$**1110$	$**1101$
$**0001$	$**0010$
$10**11$	$01**11$
$01**00$	$10**00$
$1110**$	$1101**$
$0001**$	$0010**$

Table 4.2: Patterns of strings corresponding to the vertices of truncated octahedron and octahedron ("$*$" stands for any value).

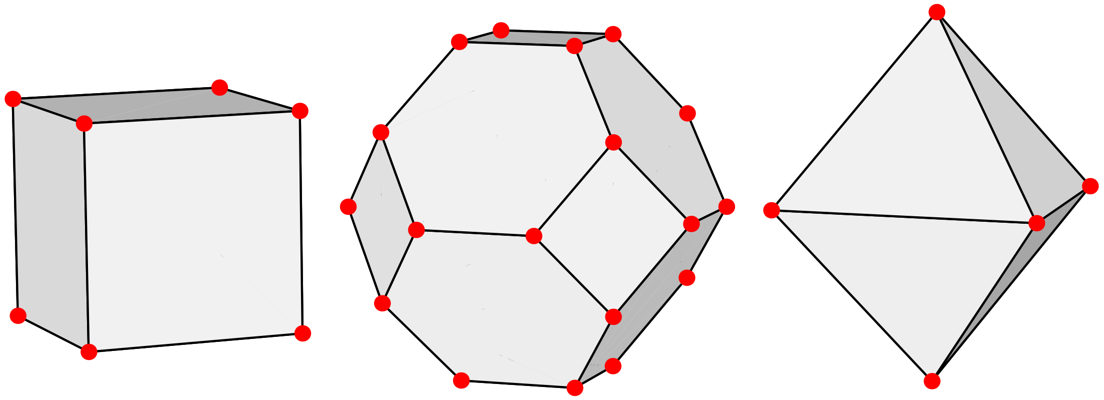

Figure 4.10: Three polyhedra (cube, truncated octahedron, and octahedron) corresponding to three different types of strings for 6 ↦ 1 QRAC with SR. The red points in Fig. 4.8 are obtained by superimposing these three polyhedra.

There is just 1 string encoded into each vertex. In this case there will be two pairs of chosen edges that belong to the same face (note the "cross" in the Fig. 4.9 formed by pairs whose arrows are pointing outwards of the page). The third pair (with the arrow pointing up) can be rotated around this face to any of the four possible positions. This corresponds to fixing four bits of the string and choosing the remaining two bits in an arbitrary way. Since the "cross" can be on any of the six faces of the cube, one can easily describe all 24 strings of this type (they are listed in the first column of Table 4.2).

- The *octahedron* has 6 vertices:

$$(\pm 1, 0, 0) \cup (0, \pm 1, 0) \cup (0, 0, \pm 1) \tag{4.21}$$

and there are 4 strings encoded into each vertex. In this case two arrows in Fig. 4.9 are pointing to opposite directions (up and down). If we fix these arrows, we can rotate the third one (pointing outwards) in any of four directions. Hence we can describe all 24 strings of this type in a similar way (see the second column of Table 4.2).

The coefficients β of the encoding states are the 64 roots of the polynomial

$$\begin{aligned}\beta^4(\beta-1)^4(4\beta^4-1)^4(36\beta^8+24\beta^4+1)^2 \\ (25\beta^8-15\beta^4+1)(400\beta^8-360\beta^4+1)(400\beta^8+56\beta^4+25).\end{aligned} \tag{4.22}$$

The obtained success probability using input randomization is

$$p = \frac{1}{2} + \frac{2+\sqrt{3}+\sqrt{15}}{16\sqrt{6}} \approx 0.694046. \tag{4.23}$$

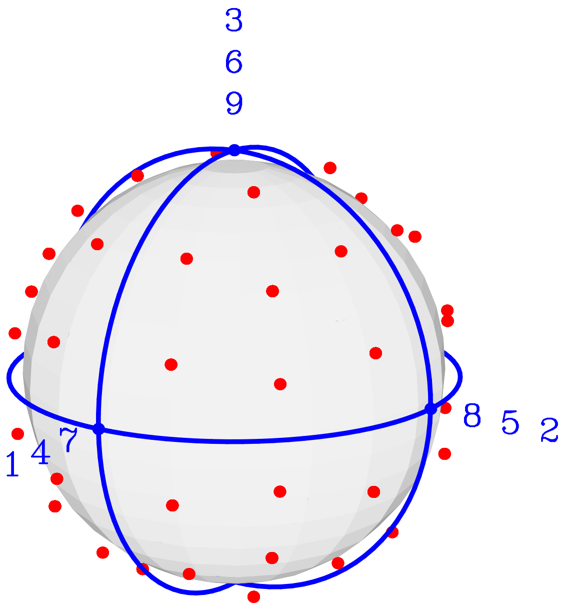

Figure 4.11: The $9 \mapsto 1$ QRAC with SR.

4.1.5 The $9 \mapsto 1$ QRAC with SR

This QRAC is a combination of three $3 \mapsto 1$ QRACs described in Sect. 3.3.2. It has three measurements along each axis:

$$\begin{aligned} \boldsymbol{v}_1 = \boldsymbol{v}_4 = \boldsymbol{v}_7 &= \pm(1,0,0), \\ \boldsymbol{v}_2 = \boldsymbol{v}_5 = \boldsymbol{v}_8 &= \pm(0,1,0), \\ \boldsymbol{v}_3 = \boldsymbol{v}_6 = \boldsymbol{v}_9 &= \pm(0,0,1). \end{aligned} \tag{4.24}$$

The measurement bases M_1, M_2, and M_3 corresponding to the Bloch vectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, and $\boldsymbol{v}_3$ are given by (3.14), (3.15), and (3.19), respectively.

The encoding points can be characterized as a $4 \times 4 \times 4$ *cubic lattice* formed by vectors (3.27) projected on the surface of the Bloch ball. Note that this lattice consists of vertices of 8 equal cubes each lying in a different octant. Then the 7 points inside of each spherical triangle in Fig. 4.11 are the projection of the vertices of the corresponding cube.[2]

All $2^9 = 512$ strings can be classified into 3 types. First consider a string $a_1a_2a_3 \in \{0,1\}^3$ and define

$$s(a_1,a_2,a_3) = \frac{|a_1-a_2|+|a_2-a_3|+|a_3-a_1|}{2}. \tag{4.25}$$

Notice that $s(a_1,a_2,a_3) \in \{0,1\}$. Now for $x = x_1x_2\dots x_9 \in \{0,1\}^9$ define

$$t(x) = s(x_1,x_4,x_7) + s(x_2,x_5,x_8) + s(x_3,x_6,x_9). \tag{4.26}$$

[2] We get 7 points instead of 8 since the projections of two diagonally opposite vertices coincide.

Then the type of the string x can be determined as follows:

$$t(x) = \begin{cases} 0,3 & \text{cube}, \\ 1 & \text{truncated cube}, \\ 2 & \text{small rhombicuboctahedron}. \end{cases} \tag{4.27}$$

These types are named after polyhedra, since each type of string is encoded into the vertices of the corresponding polyhedron (see Fig. 4.12):

- The *cube* has 8 vertices and there are 28 strings encoded into each vertex. These vertices are:
$$\frac{1}{\sqrt{3}}(\pm 1, \pm 1, \pm 1). \tag{4.28}$$
- The deformed[3] *truncated cube* has 24 vertices and there are 3 strings encoded into each vertex. These vertices are:
$$\frac{1}{\sqrt{19}}(\pm 1, \pm 3, \pm 3) \cup \frac{1}{\sqrt{19}}(\pm 3, \pm 1, \pm 3) \cup \frac{1}{\sqrt{19}}(\pm 3, \pm 3, \pm 1). \tag{4.29}$$
- The deformed[4] *small rhombicuboctahedron* also has 24 vertices and there are 9 strings encoded into each vertex. These vertices are:
$$\frac{1}{\sqrt{11}}(\pm 3, \pm 1, \pm 1) \cup \frac{1}{\sqrt{11}}(\pm 1, \pm 3, \pm 1) \cup \frac{1}{\sqrt{11}}(\pm 1, \pm 1, \pm 3). \tag{4.30}$$

The coefficients β for the corresponding qubit states $\alpha |0\rangle + \beta |1\rangle$ are the roots of the following polynomial:

$$\begin{gathered}(36\beta^8 + 24\beta^4 + 1)^{28}(1444\beta^8 + 760\beta^4 + 81)^3(484\beta^8 + 440\beta^4 + 1)^9 \\ (52128400\beta^{16} - 21509824\beta^{12} + 26780424\beta^8 - 372400\beta^4 + 15625)^3 \\ (5856400\beta^{16} - 1788864\beta^{12} + 1232264\beta^8 - 92400\beta^4 + 15625)^9. \end{gathered} \tag{4.31}$$

Using input randomization we get success probability

$$p = \frac{1}{2} + \frac{192 + 10\sqrt{3} + 9\sqrt{11} + 3\sqrt{19}}{384} \approx 0.656893. \tag{4.32}$$

4.2 Symmetric constructions

In Sect. 4.1 we have discussed in great detail $n \mapsto 1$ quantum random access codes with shared randomness for some particular values of n. Since these codes were obtained using numerical optimization, there are still some questions left open.

[3] The edges of the *truncated cube* are of the same length. In our case the eges forming triangles are $\sqrt{2}$ times longer than the other edges.

[4] The edges of the *small rhombicuboctahedron* are also of the same length, but in our case the edges forming triangles again are $\sqrt{2}$ times longer.

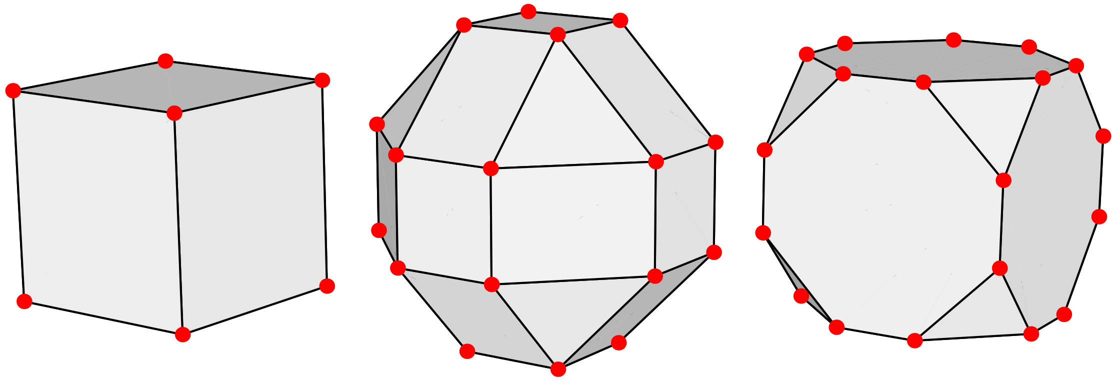

Figure 4.12: Three polyhedra (cube, small rhombicuboctahedron, and truncated cube) corresponding to three different types of strings for $9 \mapsto 1$ QRAC with SR. The red points in Fig. 4.11 are obtained by superimposing these three polyhedra.

Most importantly, are the codes for $n \geq 4$ discussed in Sect. 4.1 optimal? If this is the case, do these codes (see Figs. 4.2, 4.3, 4.4, 4.5, 4.8, and 4.11) have anything in common that makes them so good?

The purpose of this section is to shed some light on these two questions. We will explore the possibility that *symmetry* is the property that makes QRACs with SR good. In Sect. 4.2.1 we will explore what symmetries the codes found by numerical optimization have and what other symmetries are possible. In several subsequent sections we will use these symmetries to construct new codes and compare them with the numerical ones (the success probabilities of the obtained codes are summarized in Table 4.3). In Sect. 4.3 we will conclude that symmetric codes are *not* necessarily optimal and speculate about what else could potentially be used to construct good QRACs.

n	Section	Probability
4	4.2.2	0.733253
6	4.2.3	0.694042
9	4.2.4	0.656393
15	4.2.5	0.620183

Table 4.3: The success probabilities of symmetric $n \mapsto 1$ QRACs with SR. See Table 4.6 for the comparison with numerically obtained codes.

4.2.1 Symmetric great circle arrangements

If we want to construct a QRAC with SR that has some sort of symmetry, we have to choose the directions of measurements in a symmetric way. In other words, we have to symmetrically arrange the great circles that are orthogonal to the measurement directions.

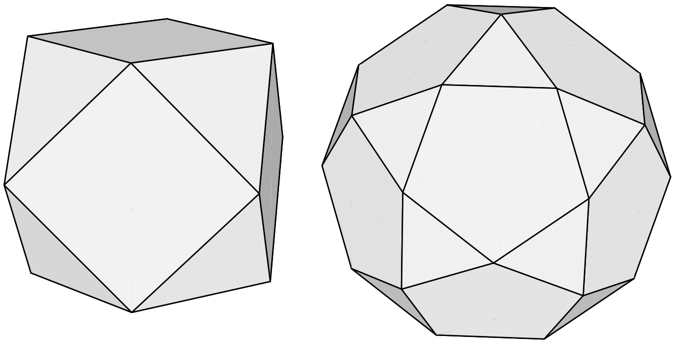

Figure 4.13: Quasiregular polyhedra: *cuboctahedron* and *icosidodecahedron*.

In this section we will discuss two ways that great circles can be arranged on a sphere in a symmetric way. These arrangements come from *quasiregular polyhedra* and *triangular symmetry groups*, respectively. The first kind of arrangement is not directly observed in numerically obtained examples, despite its high symmetry. However, the second one is observed in almost all numerically obtained codes. Since our approach is empirical, we will not justify when an arrangement is "symmetric enough"[5] to be of interest. We will use the term *symmetric codes* to refer to the codes constructed below. This is just to distinguish them from numerically obtained codes in Sect. 4.1, not because they satisfy some formal criterion of "being symmetric".

Quasiregular polyhedra

A (convex) *quasiregular polyhedron* is the intersection of a *Platonic solid* with its dual. There are only three possibilities:

$$\text{octahedron} = \text{tetrahedron} \cap \text{tetrahedron}, \tag{4.33}$$

$$\text{cuboctahedron} = \text{cube} \cap \text{octahedron}, \tag{4.34}$$

$$\text{icosidodecahedron} = \text{icosahedron} \cap \text{dodecahedron}. \tag{4.35}$$

The tetrahedron is self-dual thus the octahedron, which is the intersection of two tetrahedrons, has slightly different properties than the other two polyhedra (e.g., its all faces are equal). For this reason octahedron may be considered as a degenerate quasiregular polyhedron or not be considered quasiregular at all since it is Platonic. Thus there are only two (non-degenerate) convex quasiregular polyhedra (see Fig. 4.13).

These polyhedra have several nice properties. For example, all their edges are equivalent and there are exactly two types of faces (both regular polygons), each completely surrounded by the faces of the other type. The most relevant property

[5] Several possible criteria are: (a) any great circle can be mapped to any other by a rotation from the symmetry group of the arrangement, (b) the sphere is cut into pieces that are regular polygons, (c) the sphere is cut into pieces of the same form. However, not all examples we will give satisfy these three conditions. In fact, each condition is violated by at least one of the examples we will consider.

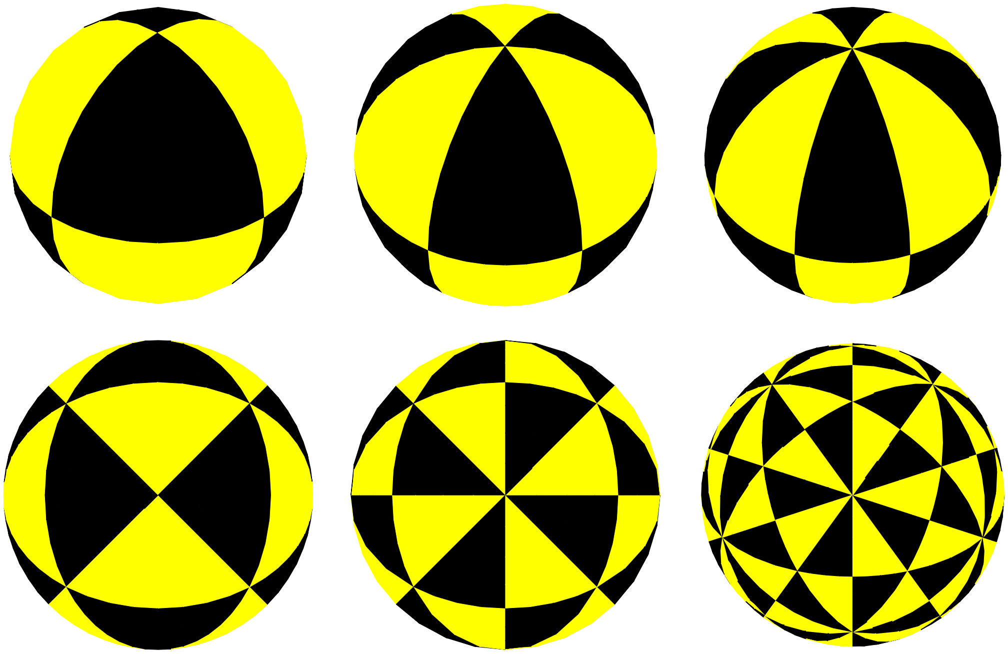

Figure 4.14: Triangular symmetry groups. First row: $(2,2,2)$, $(2,2,3)$, $(2,2,4)$. Second row: $(2,3,3)$, $(2,3,4)$, $(2,3,5)$.

for us is that their edges form great circles. Since the arrangements of great circles formed by the edges of cuboctahedron and icosidodecahedron do not appear in the numerical codes, we will use them in Sects. 4.2.2 and 4.2.3 to construct new (symmetric) $4 \mapsto 1$ and $6 \mapsto 1$ QRACs with SR, respectively.

Triangular symmetry groups

Consider a spherical triangle—it is enclosed by three planes that pass through its edges and the center of the sphere. Let us imagine that these planes are mirrors that reflect our triangle. These three reflections generate a *reflection group* [24, 25]. For some specific choices of the triangle this group is finite and the images of the triangle under different group operations do not overlap. Hence they form a *tiling* of the sphere. This tiling can also be seen as several (most likely more than three) great circles cutting the sphere into equal triangles.

We can choose any of the triangles in the tiling and repeatedly reflect it along its edges so that it moves around one of its vertices. This means that the angles of the corners that meet at any vertex of the tiling must be equal. Moreover, we do not want the triangle to intersect with any of the mirrors, so only an even number of triangles can meet at a vertex.[6]

Hence the angles of the spherical triangle must be $(\frac{\pi}{p}, \frac{\pi}{q}, \frac{\pi}{r})$ for some integers $p, q, r \geq 2$. The sum of the angles of a spherical triangle is at least π, so the numbers

[6] Fore example, if we project the edges of an icosahedron on the sphere, we obtain arcs that form a tiling with five triangles meeting at each vertex. We cannot use these arcs as mirrors, since they do not form great circles (we cannot extend any of them to a great circle, without intersecting other triangles).

p, q, r must satisfy

$$\frac{1}{p} + \frac{1}{q} + \frac{1}{r} > 1. \tag{4.36}$$

If $p \leq q \leq r$, the only solutions are: $(2, 2, k)$ for any $k \geq 2$, $(2, 3, 3)$, $(2, 3, 4)$, and $(2, 3, 5)$. The tilings corresponding to these solutions are shown in Fig. 4.14. The symmetry group of such tiling is called *triangular symmetry group* [25, pp. 158] and is denoted by (p, q, r).

We can observe these tilings in almost all numerically obtained QRACs discussed in Sect. 4.1. They are formed when the great circles corresponding to measurements partition the Bloch sphere into equal triangles. All such cases are summarized in Table 4.4. Tilings appearing in $2 \mapsto 1$ and $4 \mapsto 1$ QRACs that are not mentioned in the table can be seen as degenerate cases.

n	(p, q, r)	Polyhedron	Section and figure
3	$(2, 2, 2)$	octahedron	Sect. 4.1.1, Fig. 4.3
5	$(2, 2, 4)$	normalized octagonal dipyramid	Sect. 4.1.3, Fig. 4.5
6	$(2, 3, 3)$	normalized tetrakis hexahedron	Sect. 4.1.4, Fig. 4.8
9	$(2, 2, 2)$	octahedron	Sect. 4.1.5, Fig. 4.11

Table 4.4: Triangular symmetry groups of numerical $n \mapsto 1$ QRACs.

The tilings corresponding to triangular symmetry groups $(2, 3, 4)$ and $(2, 3, 5)$ do not appear in numerically obtained codes. Thus we will use them to construct new (symmetric) $9 \mapsto 1$ and $15 \mapsto 1$ QRACs with SR in Sects. 4.2.4 and 4.2.5, respectively. To each tiling one can associate a corresponding polyhedron with equal triangular faces. The polyhedra corresponding to tilings $(2, 3, 4)$ and $(2, 3, 5)$ are called the normalized[7] *disdyakis dodecahedron* and the normalized *disdyakis triacontahedron*, respectively.

Polyhedra arising from both types of symmetric great circle arrangements (quasiregular polyhedra and triangular symmetry groups) are summarized in Table 4.5. The great circle arrangements corresponding to the four marked polyhedra do not appear in numerically obtained codes, so we will use them to construct new (symmetric) QRACs with SR.

4.2.2 Symmetric $4 \mapsto 1$ QRAC with SR

Recall that in Sect. 3.3.3 we proved that four planes passing through the center of the Bloch sphere partition its surface into at most 14 parts. The most symmetric way to obtain 14 parts is to use the four planes parallel to the four faces of a regular *tetrahedron*. The measurements are along the four directions given by the vertices (see Fig. 4.16).

The simplest way to construct a regular tetrahedron is to choose four specific vertices of the cube, i.e., from the set $\frac{1}{\sqrt{3}}(\pm 1, \pm 1, \pm 1)$. For example, we could choose

[7] *Normalized* means that all vectors pointing from the origin to the vertices of the polyhedron are rescaled to have unit norm.

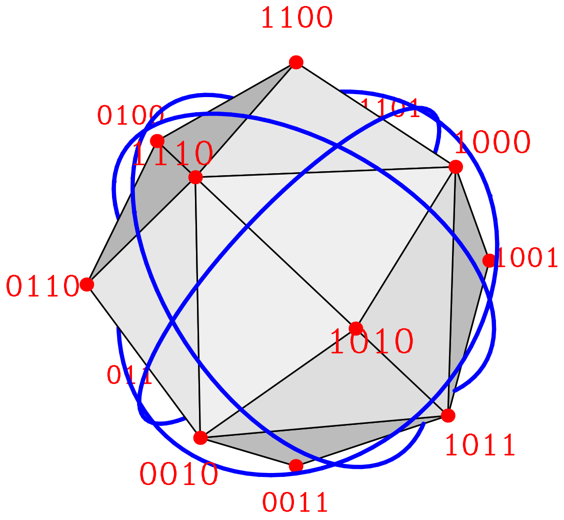

Figure 4.15: Symmetric $4 \mapsto 1$ QRAC with SR.

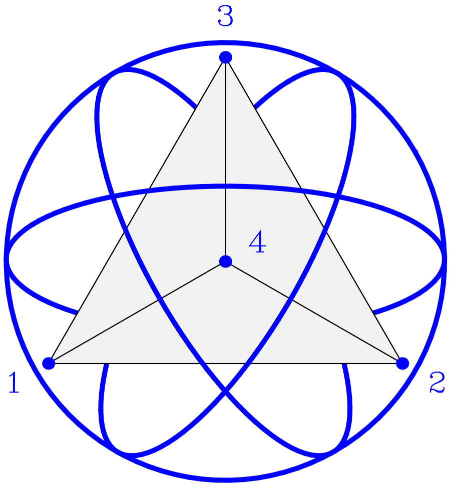

Figure 4.16: A regular tetrahedron and four great circles parallel to its faces. The circles are determined by the measurements in the direction of the vertices of the tetrahedron. The numbers at the vertices indicate the Bloch vectors of basis states $|\psi_0\rangle$ of the measurements for the $4 \mapsto 1$ QRAC shown in Fig. 4.15.

n	Faces		(p,q,r)	Polyhedron
3	8	8	$(2,2,2)$	octahedron
4	14	14	QR	cuboctahedron ✓
6	32	32	QR	icosidodecahedron ✓
6	24	32	$(2,3,3)$	normalized tetrakis hexahedron
9	48	74	$(2,3,4)$	normalized disdyakis dodecahedron ✓
15	120	212	$(2,3,5)$	normalized disdyakis triacontahedron ✓

Table 4.5: Polyhedra whose edges form great circles. The first column indicates the number of great circles. The next two indicate, respectively, the number of faces of the polyhedron and the maximal number of pieces achievable by cutting the sphere with n great circles (see Sect. 3.3.3). The fourth column indicates the triangular symmetry group (QR means quasiregular). The name of the polyhedron is given in the last column. Four marked polyhedra will be used in subsequent sections to construct symmetric QRACs with SR.

the ones with an odd number of positive coordinates. They provide us with the following pairs of antipodal Bloch vectors as the measurement bases:

$$\begin{aligned}
\boldsymbol{v}_1 &= \pm(+1,-1,-1)/\sqrt{3},\\
\boldsymbol{v}_2 &= \pm(-1,+1,-1)/\sqrt{3},\\
\boldsymbol{v}_3 &= \pm(-1,-1,+1)/\sqrt{3},\\
\boldsymbol{v}_4 &= \pm(+1,+1,+1)/\sqrt{3}.
\end{aligned} \tag{4.37}$$

The qubit states corresponding to these Bloch vectors are as follows:

$$\begin{aligned}
M_1 &= M(+1,+1),\\
M_2 &= M(+1,-1),\\
M_3 &= M(-1,+1),\\
M_4 &= M(-1,-1),
\end{aligned} \tag{4.38}$$

where

$$M(s_1,s_2) = \left\{ \frac{1}{2}\sqrt{1+\frac{s_1}{\sqrt{3}}} \begin{pmatrix} \sqrt{3}-s_1 \\ s_2(s_1-i) \end{pmatrix}, \frac{1}{2}\sqrt{1-\frac{s_1}{\sqrt{3}}} \begin{pmatrix} \sqrt{3}+s_1 \\ s_2(i-s_1) \end{pmatrix} \right\}. \tag{4.39}$$

The great circles determined by these measurements partition the Bloch ball into 14 parts. In fact, the grid formed by these circles is a projection of the edges of a *cuboctahedron* (see the part on quasireglar polyhedra in Sect. 4.2.1) on the surface of the Bloch ball (see Figs. 4.15 and 4.16).

In each of the 14 parts of the Bloch sphere a definite string can be encoded so that each bit can be recovered with a probability greater than $\frac{1}{2}$. Strange as it may seem, the remaining 2 strings ($x = 0000$ and $x = 1111$) can be encoded anywhere without affecting the success probability of this QRAC. This is not a surprise if we recall from Sect. 3.4 that the optimal encoding $\boldsymbol{r}_x$ of the string x is a unit vector in

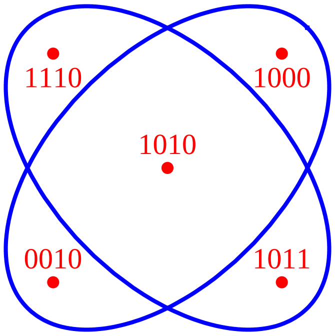

Figure 4.17: The relationship between the strings encoded into the spherical square and the adjacent spherical triangles according to the 4 ↦ 1 QRAC shown in Fig. 4.15.

the direction of $\boldsymbol{v}_x$ given by equation (3.27). In our case the Bloch vectors of the measurement bases point to the vertices of a regular tetrahedron centered at the origin. They clearly sum to zero, so $\boldsymbol{v}_{0000} = \boldsymbol{v}_{1111} = 0$. Thus the scalar product $\boldsymbol{r}_x \cdot \boldsymbol{v}_x$ in (3.26) is also zero and the success probability does not depend on the vectors $\boldsymbol{r}_{0000}$ and $\boldsymbol{r}_{1111}$. Therefore, we will ignore these two strings in the following discussion.

The other 14 strings are encoded into the vertices of a normalized *tetrakis hexahedron* (the *convex hull* of the *cube* and *octahedron*). The string $x = x_1x_2x_3x_4$ is encoded into the Bloch vector $\boldsymbol{r}(x) = \boldsymbol{r}_w(x)$, where

$$w = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \in \{0, 1\} \tag{4.40}$$

is the parity of the input. In the case $w = 0$ the encoding points are the vertices $(\pm1, 0, 0) \cup (0, \pm1, 0) \cup (0, 0, \pm1)$ of an *octahedron*:

$$\boldsymbol{r}_0(x) = (-1)^{x_4} \begin{pmatrix} 1 - |x_1 - x_4| \\ 1 - |x_2 - x_4| \\ 1 - |x_3 - x_4| \end{pmatrix}. \tag{4.41}$$

But for $w = 1$ we get the vertices $(\pm1, \pm1, \pm1)/\sqrt{3}$ of a *cube*:

$$\boldsymbol{r}_1(x) = \frac{(-1)^{x_1x_2+x_3x_4}}{\sqrt{3}} \begin{pmatrix} (-1)^{x_1+x_4} \\ (-1)^{x_2+x_4} \\ (-1)^{x_3+x_4} \end{pmatrix}. \tag{4.42}$$

Note that the Bloch vectors $\boldsymbol{r}_1(x)$ are the vertices of the same cube as the Bloch vectors of the 3 ↦ 1 QRAC discussed in Sect. 3.3.2.

One can observe the following properties of this encoding. The surface of the Bloch ball is partitioned into 6 *spherical squares* and 8 *spherical triangles*. Strings with $w = 0$ and $w = 1$ are encoded into squares and triangles, respectively. If $w = 1$ ($x = 1000$ or $x = 0111$ and their permutations), the string has one bit that differs from the other three. Such a string is encoded into the basis state of the corresponding measurement so that this bit can be recovered with certainty. If $w = 0$, the string is encoded into a square and has the following property: each of its bits takes the value that occurs more frequently at the same position in the strings of the four neighboring triangles (see Fig. 4.17 as an example).

The corresponding encoding function is $E(x) = \alpha_w |0\rangle + \beta_w |1\rangle$ with coefficients α_0, β_0 and α_1, β_1 explicitly given by

$$\begin{cases} \alpha_0 = \sqrt{\dfrac{1}{2} + (-1)^{x_4} \dfrac{1 - |x_3 - x_4|}{2}}, \\ \beta_0 = x_3 x_4 + (-1)^{x_4} \dfrac{1 - |x_1 - x_4| + i\bigl(1 - |x_2 - x_4|\bigr)}{\sqrt{2}}, \end{cases} \tag{4.43}$$

and

$$\begin{cases} \alpha_1 = \sqrt{\dfrac{1}{2} + \dfrac{s(x)}{2\sqrt{3}}}, \\ \beta_1 = (-1)^{x_3} s(x) \dfrac{(-1)^{x_1} + i(-1)^{x_2}}{\sqrt{6 + s(x) 2\sqrt{3}}}, \end{cases} \tag{4.44}$$

where $s(x) \in \{-1, 1\}$ is given by

$$s(x) = (-1)^{x_1 x_2 + x_3 x_4 + x_3 + x_4}. \tag{4.45}$$

The 14 coefficients β_0 and β_1 are the roots of the polynomial

$$\beta(\beta - 1)(4\beta^4 - 1)(36\beta^8 + 24\beta^4 + 1). \tag{4.46}$$

Using input randomization we get the same success probability for any input:

$$p = \frac{1}{2} + \frac{2 + \sqrt{3}}{16} \approx 0.733253. \tag{4.47}$$

It is surprising that despite higher symmetry (compare Figs. 4.4 and 4.15) this QRAC has a lower success probability than the $4 \mapsto 1$ QRAC discussed in Sect. 4.1.2.

4.2.3 Symmetric $6 \mapsto 1$ QRAC with SR

According to the discussion in Sect. 3.3.3, six great circles can cut the sphere into at most 32 parts. It turns out that there is a very symmetric arrangement that achieves this maximum. Observe that the *dodecahedron* has 12 faces and diametrically opposite ones are parallel. For each pair of parallel faces we can draw a plane through the origin parallel to both faces. These six planes intersect the sphere in six great circles that define our measurements. They are the projections of the edges of the *icosidodecahedron* (see Fig. 4.13), which is one of the quasiregular polyhedra discussed in Sect 4.2.1.

There is another way to describe these measurements. Notice that the *icosahedron* (the dual of the dodecahedron) has 12 vertices that consist of six antipodal

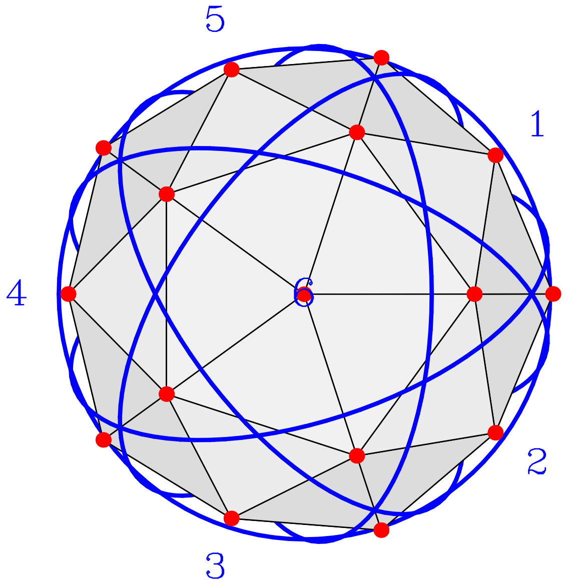

Figure 4.18: Symmetric $6 \mapsto 1$ QRAC with SR.

pairs. Our measurements are along the six directions defined by these pairs. The coordinates of the vertices of the icosahedron are as follows:

$$\frac{1}{\sqrt{1+\tau^2}}(0,\pm\tau,\pm1) \cup \frac{1}{\sqrt{1+\tau^2}}(\pm1,0,\pm\tau) \cup \frac{1}{\sqrt{1+\tau^2}}(\pm\tau,\pm1,0), \tag{4.48}$$

where $\tau = \frac{1+\sqrt{5}}{2}$ is the *golden ratio* (the positive root of $x^2 = x + 1$).

Each of the 64 strings is encoded either in a vertex of an icosahedron or dodecahedron. They have 12 and 20 vertices, respectively, so there are two strings encoded in each vertex. The union of the icosahedron and the dodecahedron is called the *pentakis dodecahedron* (see the polyhedron in Fig. 4.18).

The success probability of this code is

$$p = \frac{1}{2} + \frac{\sqrt{5}}{32} + \frac{1}{96}\sqrt{75 + 30\sqrt{5}} \approx 0.694042. \tag{4.49}$$

4.2.4 Symmetric $9 \mapsto 1$ QRAC with SR

This code is based on the triangular tiling of the sphere whose symmetry group is $(2, 3, 4)$. The great circles corresponding to measurements coincide with the projection of the edges of the *normalized disdyakis dodecahedron*. We can think of this QRAC as the union of $3 \mapsto 1$ and $6 \mapsto 1$ codes. The first three measurements are along the coordinate axis as in the $3 \mapsto 1$ QRAC discussed in Sect. 3.3.2. The remaining six measurements are exactly the same as for the $6 \mapsto 1$ code discussed in Sect. 4.1.4 (see Figs. 4.7 and 4.8), i.e., they are along the six antipodal pairs of 12 vertices of the cuboctahedron shown in Fig. 4.13. Note that a great circle

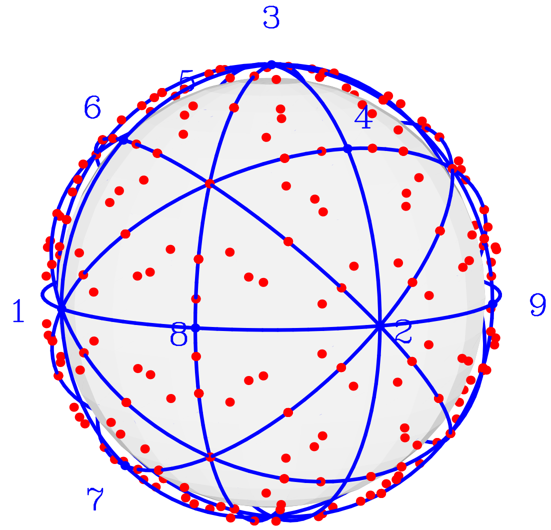

Figure 4.19: Symmetric $9 \mapsto 1$ QRAC with SR.

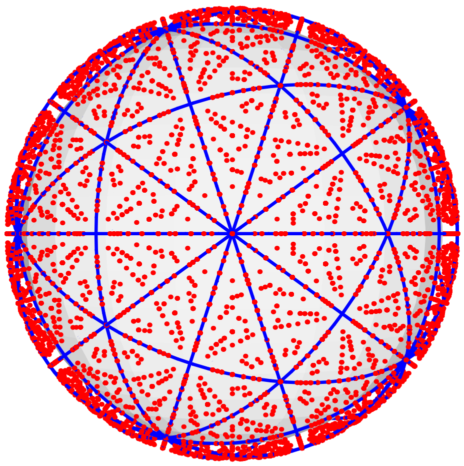

Figure 4.20: Symmetric $15 \mapsto 1$ QRAC with SR.

of the first kind cannot be transformed to a great circle of the second kind via an operation from the symmetry group of the code.[8]

The resulting QRAC is shown in Fig. 4.19 and its success probability is

$$p \approx 0.656393. \tag{4.50}$$

4.2.5 Symmetric $15 \mapsto 1$ QRAC with SR

The triangular symmetry group of this code is $(2, 3, 5)$ and the great circles coincide with the projection of the edges of the *normalized disdyakis triacontahedron*. To understand what the measurements are in this case, note that the *icosidodecahedron* (see Fig. 4.13) has 30 vertices. Their coordinates are:

$$(\pm 1, 0, 0) \cup (0, \pm 1, 0) \cup (0, 0, \pm 1), \tag{4.51}$$

$$\frac{1}{2\tau}(\pm 1, \pm\tau, \pm\tau^2) \cup \frac{1}{2\tau}(\pm\tau^2, \pm 1, \pm\tau) \cup \frac{1}{2\tau}(\pm\tau, \pm\tau^2, \pm 1). \tag{4.52}$$

The measurement directions are given by 15 antipodal pairs of these vertices.

The obtained QRAC is shown in Fig. 4.20. Its success probability is

$$p \approx 0.620183. \tag{4.53}$$

4.3 Discussion

In this section we will compare and analyze the numerical and symmetric QRACs with SR described in Sects. 4.1 and 4.2, respectively. Hopefully these observations

[8] For the other three symmetric codes we can transform any circle to any other in this way, i.e., the symmetry group acts transitively on the circles.

can be used to find new $n \mapsto 1$ QRACs with SR or to generalize the existing ones (see Sect. 5.3 for possible generalizations).

The success probabilities of numerical and symmetric QRACs with SR are given in Tables 4.1 and 4.3, respectively (see Table 4.6 for the comparison). We see that none of the symmetric codes discussed in Sect. 4.2 is optimal. However, the success probabilities of numerical and symmetric codes do not differ much. Moreover, recall that there are two more symmetric codes ($3 \mapsto 1$ and $6 \mapsto 1$) that coincide with the numerically obtained ones (see Table 4.5). Concerning these two codes we can reach more optimistic conclusions: the $3 \mapsto 1$ QRAC is optimal (see Sect. 3.6) and possibly the $6 \mapsto 1$ QRAC (see Sect. 4.1.4) is as well, since we did not manage to improve it in Sect. 4.2.3.

n	Section	Probability
4	4.1.2	0.741481
	4.2.2	> 0.733253
6	4.1.4	0.694046
	4.2.3	> 0.694042
9	4.1.5	0.656893
	4.2.4	> 0.656393
15		0.620355
	4.2.5	> 0.620183

Table 4.6: Comparison of the success probabilities of $n \mapsto 1$ QRACs with SR. For each n the first probability corresponds to a numerical code, but the second one to a symmetric code. For $n = 15$ we do not have numerical results, so we just use five measurements along each coordinate. In fact, the numerical $4 \mapsto 1$ and $9 \mapsto 1$ QRACs also use measurements only along coordinate axis. The $6 \mapsto 1$ QRAC with two measurements along each coordinate axis has success probability 0.686973.

We just saw that symmetric QRACs are not necessarily optimal. One could ask if there are other heuristic methods that potentially could be used to construct good QRACs with SR. We will give a few speculations in the remainder of this section. In particular, we will discuss some special kinds of measurements that could be useful. To make the discussion more general, we will not restrict ourselves to the case of a single qubit.

Definition. Two orthonormal bases $\mathcal{B}_1$ and $\mathcal{B}_2$ of $\mathbb{C}^d$ are called *mutually unbiased bases* (MUBs) if $|\langle\psi_1|\psi_2\rangle|^2 = \frac{1}{d}$ for all $|\psi_1\rangle \in \mathcal{B}_1$ and $|\psi_2\rangle \in \mathcal{B}_2$. The maximal number of pairwise mutually unbiased bases in $\mathbb{C}^d$ is $d + 1$. [26]

When $d = 2$, equation (3.7) implies that Bloch vectors corresponding to basis vectors of *different* mutually unbiased bases are orthogonal[9]. There are three such

[9] The notion of the Bloch vector can be generalized for $d \geq 2$ (see [28]). Then a similar duality holds as well (see equation (5.4) in Sect. 5.3): mutually unbiased quantum states correspond to orthogonal Bloch vectors, but orthogonal quantum states correspond to "mutually unbiased" Bloch vectors, i.e., equiangular vectors pointing to the vertices of a regular simplex.

bases in $\mathbb{C}^2$ and their Bloch vectors correspond to the vertices of an octahedron. For example, the bases M_1, M_2, and M_3 defined in Sects. 3.3.1 and 3.3.2 are MUBs (they correspond to measuring along x, y, and z axis).

Note that the measurements for numerical $2 \mapsto 1$, $3 \mapsto 1$, $4 \mapsto 1$, and $9 \mapsto 1$ QRACs are performed entirely using MUBs and three out of five measurement bases for numerical $5 \mapsto 1$ QRAC are also MUBs.

There is another very special measurement that appears in our QRACs.

Definition. A set of d^2 unit vectors $|\psi_i\rangle \in \mathbb{C}^d$ is called *symmetric, informationally complete POVM* (SIC-POVM) if $|\langle\psi_i|\psi_j\rangle|^2 = \frac{1}{d+1}$ for any i, j. [27]

For $d = 2$ there are four such quantum states. Again, from equation (3.7) we see that the inner product between any two Bloch vectors corresponding to these states is $-\frac{1}{3}$. Such equiangular Bloch vectors are exactly the vertices of a tetrahedron, e.g., $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\boldsymbol{v}_3$, $\boldsymbol{v}_4$ defined in (4.37). They were used in Sect. 4.2.2 to construct a symmetric $4 \mapsto 1$ QRAC.

Let us compare numerical and symmetric $4 \mapsto 1$ QRACs from Sects. 4.1.2 and 4.2.2, respectively. The first one is based on MUBs and is not very symmetric. Moreover, it looks like we are wasting one out of four bits, since two measurements are along the same direction. However, all measurement directions in the Bloch sphere are mutually orthogonal, except the ones that coincide. The second $4 \mapsto 1$ code is based on a SIC-POVM and is very symmetric. However, it appears that in this case we are wasting two out of 16 strings, since the way we encode them does not influence the success probability.

Now, if we compare the success probabilities of both $4 \mapsto 1$ codes (see Table 4.6), we see that the first one is clearly better. Hence we conclude that

orthogonality of the measurement Bloch vectors
seems to be more important than *symmetry*.

One can come to a similar conclusion when comparing $9 \mapsto 1$ and $15 \mapsto 1$ codes. Thus it looks like using roughly $n/3$ measurements along each coordinate axis is quite a good heuristic for constructing $n \mapsto 1$ QRAC with SR (see Sect. 3.8.2).

Chapter 5

Conclusion

5.1 Summary

We study the *worst* case success probability of random access codes with shared randomness. Yao's principle (see equation (2.5) in Sect. 2.2) and input randomization (see Theorem 1) is applied to consider the *average* case success probability instead (this works in both classical and quantum cases).

In Sect. 2.3.2 we construct an optimal *classical* $n \mapsto 1$ RAC with SR as follows (see Theorem 2): Alice XORs the input string with n random bits she shares with Bob, computes the majority and sends it to Bob; if the ith bit is requested, Bob outputs the ith bit of the shared random string XORed with the received bit. The asymptotic success probability of this code is given by equation (2.27) in Sect. 2.3.2:

$$p(n) \approx \frac{1}{2} + \frac{1}{\sqrt{2\pi n}}. \tag{5.1}$$

The worst case success probability of an optimal *quantum* RAC with SR satisfies the following inequalities:

$$\frac{1}{2} + \sqrt{\frac{2}{3\pi n}} \leq p(n) \leq \frac{1}{2} + \frac{1}{2\sqrt{n}}. \tag{5.2}$$

These upper and lower bounds are obtained in Sects. 3.6 and 3.8.1, respectively.

The success probabilities of classical and quantum RACs with SR are compared in Fig. 5.1.

5.2 Open problems for $n \mapsto 1$ QRACs

Lower bound by orthogonal measurements. The known $2 \mapsto 1$ and $3 \mapsto 1$ QRACs (see Sect. 3.3) and our numerical $4 \mapsto 1$ and $9 \mapsto 1$ QRACs with SR (see Sects. 4.1.2 and 4.1.5) suggest that MUBs can be used to obtain good QRACs (see Sect. 4.3). Indeed, $n \mapsto 1$ QRAC with orthogonal measurements (see Sect. 3.8.2) is better than the one with random measurements (see Sect. 3.8.1). However, we were not able

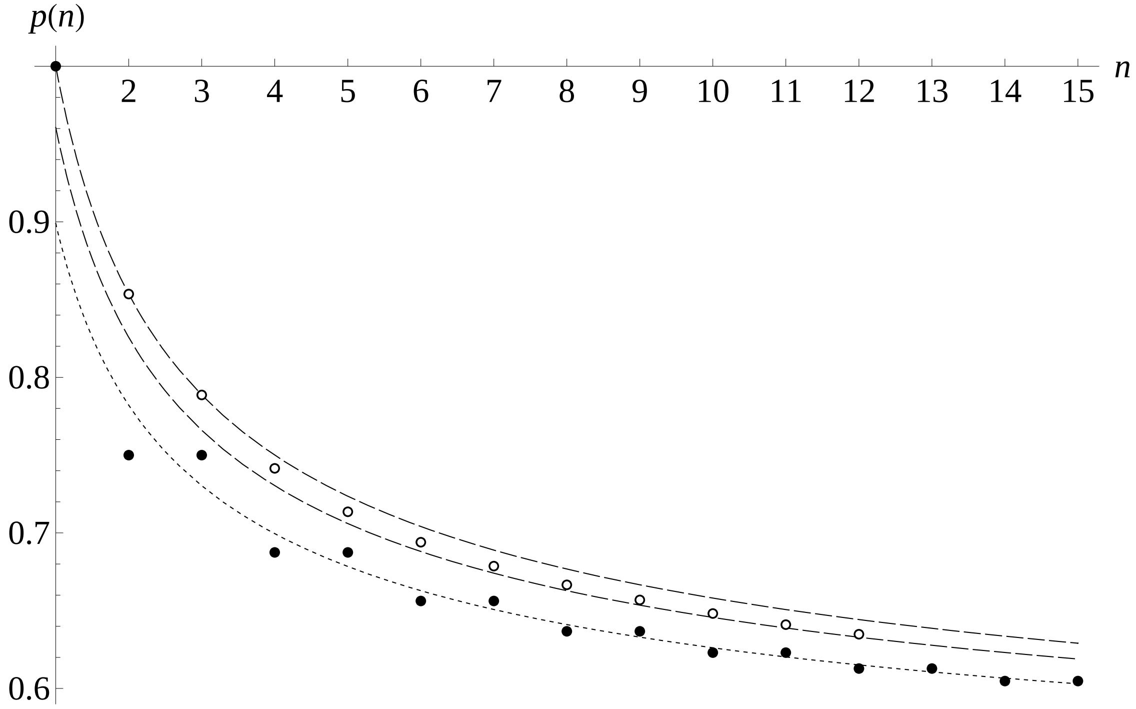

Figure 5.1: Comparison of success probabilities of classical and quantum RACs. Black dots correspond to optimal classical RACs and the dotted line shows the asymptotic behavior. Circles correspond to numerical QRACs and dashed lines to quantum upper and lower bounds, respectively.

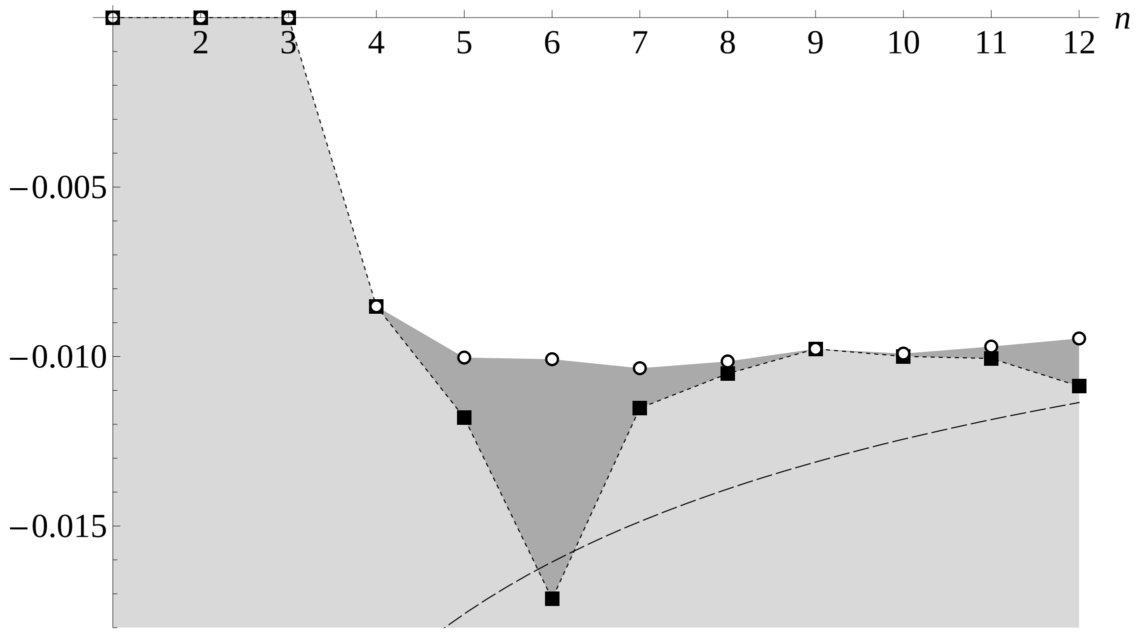

Figure 5.2: Close-up of the narrow region in Fig. 5.1 between the quantum upper and lower bound (everything is shown relative to the upper bound that corresponds to the horizontal axis). Circles indicate the gap between the upper bound and numerical QRACs with SR. Black squares show the gap between the upper bound and the lower bound by measurements along coordinate axes (see Fig. 3.9). Dashed line corresponds to the gap between the quantum upper bound and the lower bound by random measurements.

to obtain an asymptotic expression for its success probability. This is equivalent to obtaining an asymptotic expression for (3.55), i.e., the average distance traveled by a random walk with roughly $n/3$ steps along each coordinate axis.

In Fig. 5.2 we show how close both lower bounds and the success probabilities of numerical QRACs are relative to the upper bound from Sect. 3.6. Assume that Alice and Bob are given a point in the light gray region in Fig. 5.2 and asked to construct a QRAC with SR whose success probability is at least as good. Then they can use measurements along coordinate axis as in Sect. 3.8.2. If the point is in the dark gray region, they can use one of the numerical codes from Sect. 4.1. However, if it is in the white region, they have to solve the next open problem.

Optimality of numerical codes. Prove the optimality of any of the numerically obtained $n \mapsto 1$ QRACs with SR for $n \geq 4$ discussed in Sect. 4.1. Are the optimal constructions unique (up to isomorphism)?

Prove the "Homer conjecture" that quantum RACs with SR are at least as good as their classical counterparts in the sense discussed at the end of Sect. 3.4.

5.3 Possible generalizations

There are several ways that random access codes with SR can be generalized, both classically and quantumly. In particular, one can consider

- $n \overset{p}{\mapsto} 1$ codes in base d, $d > 2$ (called *qudits* in the quantum case),
- $n \overset{p}{\mapsto} m$ codes with $m > 1$,
- $n \overset{p}{\mapsto} m$ codes where any $k > 1$ bits (qubits) must be recovered.

Of course, one can consider several of these generalizations simultaneously. In the setting without shared randomness such generalizations have already appeared in the literature (see Sect. 1.2). We will briefly introduce the notion of the generalized Bloch vector which we believe can be useful to study such generalizations (it has been explicitly used in [5] to prove the impossibility of $2^m \overset{p}{\mapsto} m$ QRAC with $p > 1/2$, when SR is not allowed).

The notion of the Bloch vector introduced in Sect. 3.1.1 can be generalized for $d > 2$. For example, to write down the density matrix for $d = 3$ one uses eight *Gell-Mann matrices* denoted by λ_i instead of three Pauli matrices σ_i defined in equation (3.5). In general one needs $d^2 - 1$ matrices λ_i that span the set of all traceless $d \times d$ Hermitian matrices. A convenient choice of λ_i are the so called *generalized Gell-Mann matrices*, also known as the *generators of the Lie algebra of* $SU(d)$, given in [31]. We can use them to generalize equation (3.6):

$$\rho = \frac{1}{d}\left(I + \sqrt{\frac{d(d-1)}{2}}\,\boldsymbol{r}\cdot\boldsymbol{\lambda}\right), \tag{5.3}$$

where $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_{d^2-1})$ and $\boldsymbol{r} \in \mathbb{R}^{d^2-1}$ is the *generalized Bloch vector*[1] or *coherence vector* [28, 30]. Since the λ_i are chosen so that $\operatorname{Tr}\lambda_i = 0$ and $\operatorname{Tr}(\lambda_i\lambda_j) = 2\delta_{ij}$, equation (3.7) generalizes to

$$\left|\langle\psi_1|\psi_2\rangle\right|^2 = \operatorname{Tr}(\rho_1\rho_2) = \frac{1}{d}\big(1 + (d-1)\,\boldsymbol{r}_1\cdot\boldsymbol{r}_2\big). \tag{5.4}$$

If we want to recover a base d digit, we perform a measurement in an orthonormal basis $\{|\psi_1\rangle, \dots, |\psi_d\rangle\}$ of $\mathbb{C}^d$. Since $|\langle\psi_i|\psi_j\rangle|^2 = 0$ for any pair $i \neq j$, the corresponding Bloch vectors must satisfy $\boldsymbol{r}_i\cdot\boldsymbol{r}_j = -\frac{1}{d-1}$. This means that they are the vertices of a regular simplex that belongs to a $(d-1)$-dimensional subspace and is centered at the origin (for $d = 2$ this is just a line segment).

On the other hand, in Sect. 4.3 we observed that it might be advantageous to perform measurements along orthogonal directions in the Bloch sphere to recover different bits. Let $\boldsymbol{r}_i \perp \boldsymbol{s}_j$ be two orthogonal Bloch vectors. Then the corresponding quantum states $|\psi_i\rangle$ and $|\varphi_j\rangle$ must satisfy $|\langle\psi_i|\varphi_j\rangle|^2 = \frac{1}{d}$. This is exactly the

[1] Our normalization follows [30], where the generalized Bloch sphere has radius 1. Another widely used convention is to assume radius $\sqrt{2(d-1)/d}$, e.g., see [28, 29].

case when $|\psi_i\rangle$ and $|\varphi_j\rangle$ belong to *different* mutually unbiased bases (see Sect. 4.3). This suggests that distinct bits should be recovered using mutually unbiased measurements. Note that the Bloch vectors of the states from two MUBs correspond to the vertices of two regular simplices in mutually orthogonal subspaces. In general, the Bloch vectors of the states from all $d+1$ MUBs are the vectices of the so called *complementarity polytope* [32], which is just the octahedron when $d = 2$.

The conclusion of Sect. 4.3 and our discussion above suggests the use of MUBs to construct QRACs also for $d > 2$. Such attempts have already been made [6, 8]. Galvão [6] gives an example of $2 \overset{0.79}{\longmapsto} 1$ QRAC for qutrits ($d = 3$) and Casaccino et al. [8] numerically investigate $(d + 1) \mapsto 1$ QRACs based on MUBs for d-level quantum systems. However, there is a significant difference between the qubit and qudit case. Recall that for $d = 2$ the optimal way to encode the message x is to use a unit vector in the direction of $\boldsymbol{v}_x$ (see equation (3.27) in Sect. 3.4). A similar expression for $\boldsymbol{v}_x$ can be obtained when $d > 2$, but then the matrix ρ assigned to $\boldsymbol{r} = \boldsymbol{v}_x / \|\boldsymbol{v}_x\|$ according to equation (5.3) is not necessarily *positive semidefinite* and hence may not be a valid density matrix. However, it is known that for small enough values of $\|\boldsymbol{r}\|$ (in our case[1] $\|\boldsymbol{r}\| \leq \frac{1}{d-1}$), *all* Bloch vectors correspond to valid density matrices [29]. Hence, if we cannot use the pure state corresponding to $\boldsymbol{v}_x / \|\boldsymbol{v}_x\|$, we can always use the mixed state corresponding to $\frac{1}{d-1}\boldsymbol{v}_x / \|\boldsymbol{v}_x\|$. If one knows more about the shape of the region corresponding to valid quantum states, one can make a better choice and use a longer vector, possibly in a slightly different direction. Unfortunately, apart from convexity, not much is known about this shape. Already for $d = 3$ it is rather involved [28, 29]. In general the conditions (in terms of the coordinates of the generalized Bloch vector $\boldsymbol{r}$) for ρ to have non-negative eigenvalues are given in [30, 28].

However, for proving only an upper bound, one can ignore all such details. Thus we believe it might be possible to generalize our upper bound (see Sect. 3.6 and 3.7) using generalized Bloch vectors. It would be interesting to compare such a result with the upper bound (3.46) that was obtained by Ben-Aroya et al. in [16].

Finally, another way of generalizing QRACs with SR is to add other resources. A good candidate is *shared entanglement.*

Appendices

Appendix A

Combinatorial interpretation of sums

In this appendix we give a combinatorial interpretation of the sums in equations (2.22) and (2.23) from Sect. 2.3.2. This interpretation is formalized in the form of equations (A.1) and (A.2). We referred to these equations in Sect. 2.3.2 to obtain an exact formula (2.26) for the average success probability of an optimal classical RAC.

Let us consider a set of n distinct elements and count *the number of ways to choose more than half of n elements and mark one of them as special*. There are two approaches: first choose the elements and then mark the special one or first choose the special one and then choose the others.

In the first scenario there are $i\binom{n}{i}$ ways to choose exactly i elements and mark one of them as special. If we have to choose more than half, we obtain the sum $\sum_{i=m+1}^{n} i\binom{n}{i}$ where $m = \left\lfloor \frac{n}{2} \right\rfloor$.

In the second scenario there are n ways to choose the special element. Then there are $l = n - 1$ elements left and at least m of them must be taken to have more than half of n elements in total. The number of ways to do this corresponds to the number of subsets of size at least m of a set of l distinct elements. Let us consider the cases when l is odd and even separately.

If $n = 2m$ then $l = 2m - 1$ is odd. To each "large" subset of size i ($m \leq i \leq l$) we can assign a unique "small" subset (the complement set) of size $l-i$ ($0 \leq l-i \leq m-1$), and vice versa. Each subset is either "large" or "small", so the number of "large" and "small" subsets is the same—it is half of the number of all subsets, i.e., $2^l/2 = 2^{2m-2}$.

If $n = 2m+1$ then $l = 2m$ is even. The "large" subsets have $m+1 \leq i \leq l$ elements, but the "small" ones: $0 \leq l-i \leq m-1$. Let us call the remaining $\binom{2m}{m}$ subsets of size m "balanced". In this case the bijection between the "large" and "small" subsets holds as well, but it maps the "balanced" subsets to themselves. Thus the total number of all subsets is "large" + "small" + $\binom{2m}{m} = 2^l$. The number of "large" subsets is $\left(2^l + \binom{2m}{m}\right)/2 = 2^{2m-1} + \frac{1}{2}\binom{2m}{m}$.

Both counting methods must give the same results, so for odd and even n we obtain, respectively:

$$\sum_{i=m+1}^{2m+1} i\binom{2m+1}{i} = (2m+1)\cdot\left(2^{2m-1}+\frac{1}{2}\binom{2m}{m}\right), \tag{A.1}$$

$$\sum_{i=m+1}^{2m} i\binom{2m}{i} = 2m\cdot 2^{2m-2}. \tag{A.2}$$

We would like to acknowledge Juris Smotrovs for providing this interpretation.

Appendix B

POVMs versus orthogonal measurements

An orthogonal (or *von Neumann*) measurement is not the most general type of measurement of a quantum system. In general a POVM measurement [33, 34] may extract more information. In this appendix we show that in the qubit case POVMs can be simulated using a probabilistic combination of *enhanced orthogonal measurements*, as defined in Sect. 3.7 (such a measurement is either an orthogonal measurement or a constant function). To define a POVM we have to introduce the notion of a positive semidefinite matrix [35].

Definition. A complex square matrix E is called *positive semidefinite* if $\langle\psi| E |\psi\rangle \geq 0$ for all $|\psi\rangle$.

An equivalent definition is that E is diagonalizable and all eigenvalues of E are real and non-negative. Thus E is Hermitian.

Definition. A *positive operator-valued measure* (POVM) is a set $\{E_1, \dots, E_m\}$ of positive semidefinite matrices such that $\sum_{i=1}^{m} E_i = I$. [33, 34]

POVM measurements can have an arbitrary number of outcomes, but in the case of $n \mapsto 1$ QRACs we have to consider only two-outcome single-qubit POVMs. Such a POVM can be specified by $\{E_0, E_1\}$, where E_0 is positive semidefinite and $E_1 = I - E_0$. Since E_0 is also Hermitian, we can find a basis $\mathcal{B} = \{|\psi_0\rangle, |\psi_1\rangle\}$ in which E_0 is diagonal, i.e., $E_0 = \left(\begin{smallmatrix} a & 0 \\ 0 & b \end{smallmatrix}\right)$. In this basis $E_1 = \left(\begin{smallmatrix} 1-a & 0 \\ 0 & 1-b \end{smallmatrix}\right)$. Since both E_0 and E_1 are positive semidefinite, $0 \leq a \leq 1$ and $0 \leq b \leq 1$. An arbitrary pure qubit state $|\psi\rangle$ in the basis $\mathcal{B}$ can be specified by (3.1). When $|\psi\rangle$ is measured, the probabilities of the outcomes are

$$\begin{cases} P_0 = \langle\psi| E_0 |\psi\rangle = a\cos^2\dfrac{\theta}{2} + b\sin^2\dfrac{\theta}{2}, \\ P_1 = \langle\psi| E_1 |\psi\rangle = (1-a)\cos^2\dfrac{\theta}{2} + (1-b)\sin^2\dfrac{\theta}{2}. \end{cases} \tag{B.1}$$

Let us consider the following process (see Fig. B.1) that simulates the POVM measurement $\{E_0, E_1\}$:

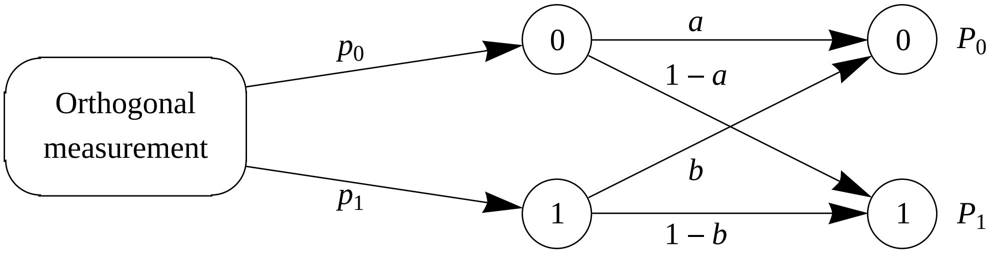

Figure B.1: A simulation of the POVM measurement $\{E_0, E_1\}$ on a qubit using an orthogonal measurement and a post-processing of the measurement result.

1. perform an orthogonal measurement in the basis $\mathcal{B} = \{|\psi_0\rangle, |\psi_1\rangle\}$,
2. perform the following post-processing of the outcome of the measurement:
   - *if the outcome was* 0: output 0 with probability a and output 1 with probability $1-a$,
   - *if the outcome was* 1: output 0 with probability b and output 1 with probability $1-b$.

To see why this process is equivalent to the POVM measurement $\{E_0, E_1\}$, consider a pure qubit state $|\psi\rangle$ given by (3.1) in the basis $\mathcal{B}$. When $|\psi\rangle$ is measured in the basis $\mathcal{B} = \{|\psi_0\rangle, |\psi_1\rangle\}$, the probabilities of the outcomes 0 and 1 are as follows (see also equation (3.8) in Sect. 3.1.1):

$$\begin{cases} p_0 = |\langle\psi_0|\psi\rangle|^2 = \cos^2\dfrac{\theta}{2}, \\ p_1 = |\langle\psi_1|\psi\rangle|^2 = \sin^2\dfrac{\theta}{2}. \end{cases} \tag{B.2}$$

Now it is simple to verify that the process shown in Fig. B.1 has the same outcome probabilities (B.1) as the POVM measurement. However, this process is not a probabilistic combination of enhanced orthogonal measurements, since it involves a probabilistic post-processing of the measurement result. To obtain the desired result, we have to modify it. The key idea is that with a certain probability the output can be produced without performing an actual measurement.

Let $\mu = \min\{a, b\}$. Whatever state is input to the process shown in Fig. B.1, the probability P_0 to output 0 is at least μ, because

$$P_0 = p_0 a + p_1 b \geq (p_0 + p_1)\mu = \mu. \tag{B.3}$$

Note that μ does not depend on the state being measured. This means that one can output 0 with probability μ without performing an actual measurement. A similar lower bound holds for P_1 as well:

$$P_1 = p_0(1-a) + p_1(1-b) \geq (p_0 + p_1)(1-M) = 1-M, \tag{B.4}$$

where $M = \max\{a, b\} = a + b - \mu$. Let us consider the following probabilistic combination of four decoding strategies:

- *with probability* c_0: output 0 without performing a measurement,
- *with probability* c_1: output 1 without performing a measurement,
- *with probability* c_{01}: measure in the basis $\{|\psi_0\rangle, |\psi_1\rangle\}$,
- *with probability* c_{10}: measure in the opposite basis $\{|\psi_1\rangle, |\psi_0\rangle\}$.

The resulting probabilities of outcomes for this process are

$$\begin{cases} P_0 = c_0 + c_{01}p_0 + c_{10}p_1, \\ P_1 = c_1 + c_{01}p_1 + c_{10}p_0. \end{cases} \tag{B.5}$$

We can use the lower bounds (B.3) and (B.4) for P_0 and P_1, respectively, to assign the probabilities c_0, c_1, c_{01}, and c_{10} in the following way:

$$\begin{cases} c_0 = \mu, \\ c_1 = 1 - (a + b) + \mu, \\ c_{01} = a - \mu, \\ c_{10} = b - \mu \end{cases} \tag{B.6}$$

(note that at least one of the probabilities c_{01} or c_{10} will be zero). It is not hard to verify that after the assignment (B.6) the probabilities P_0 and P_1 in (B.5) will match the probabilities of outcomes (B.1) of the POVM measurement.

Thus for each qubit POVM given by a and b one can find a probabilistic combination of enhanced orthogonal measurements given by c_0, c_1, c_{01}, and c_{10}, such that in both cases the probabilities of outcomes are the same.

Example. For $a = b = 1/2$ we have $c_0 = c_1 = 1/2$ and $c_{01} = c_{10} = 0$, corresponding to random guessing (observe that $E_0 = E_1$ in this case).

Example. However, $a = 1$ and $b = 0$ corresponds to a projective measurement in basis $\{|\psi_0\rangle, |\psi_1\rangle\}$, because $c_{01} = 1$ and $c_{10} = c_0 = c_1 = 0$.

Example. Finally, $a = 1$ and $b = 1$ corresponds to a constant function 0, because $c_0 = 1$ and $c_{01} = c_{10} = c_1 = 0$.

References

[1] Stephen Wiesner, “Conjugate coding,” *SIGACT News*, vol. 15, issue 1, pp. 78–88, 1983.

[2] Andris Ambainis, Ashwin Nayak, Amnon Ta-Shma, Umesh Vazirani, “Dense quantum coding and a lower bound for 1-way quantum automata,” Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC’99), pp. 376–383, 1999. `arXiv:quant-ph/9804043v2`

[3] Andris Ambainis, Ashwin Nayak, Amnon Ta-Shma, Umesh Vazirani, “Dense Quantum Coding and Quantum Finite Automata,” *Journal of the ACM*, vol. 49, no. 4, pp. 496–511, 2002.

[4] Ashwin Nayak, “Optimal lower bounds for quantum automata and random access codes,” Proceedings of the 40th IEEE Symposium on Foundations of Computer Science (FOCS’99), pp. 369–376, 1999. `arXiv:quant-ph/9904093v3`

[5] Masahito Hayashi, Kazuo Iwama, Harumichi Nishimura, Rudy Raymond, Shigeru Yamashita, “$(4, 1)$-Quantum Random Access Coding Does Not Exist,” *New J. Phys.*, vol. 8, no. 8, pp. 129, 2006. `arXiv:quant-ph/0604061v1`

[6] Ernesto F. Galvão, “Foundations of quantum theory and quantum information applications,” PhD thesis, University of Oxford, 2002.

[7] Robert W. Spekkens, Daniel H. Buzacott, Anthony J. Keehn, Ben Toner, Geoff J. Pryde, “Preparation contextuality powers parity-oblivious multiplexing,” *Phys. Rev. Lett.*, vol. 102, no. 1, 010401, 2009. `arXiv:0805.1463v2`

[8] Andrea Casaccino, Ernesto F. Galvão, Simone Severini, “Extrema of discrete Wigner functions and applications,” *Phys. Rev. A* **78**, 022310, 2008. `arXiv:0805.3466v2`

[9] Hartmut Klauck, “Lower bounds for quantum communication complexity,” Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS’01), pp. 288, 2001. `arXiv:quant-ph/0106160v3`

[10] Scott Aaronson, “Limitations of Quantum Advice and One-Way Communication,” Proceedings of the 19th Annual IEEE Conference on Computational Complexity (CCC’04), pp. 320–332, 2004. `arXiv:quant-ph/0402095v4`

[11] Dmitry Gavinsky, Julia Kempe, Oded Regev, Ronald de Wolf, "Bounded-Error Quantum State Identification and Exponential Separations in Communication Complexity," Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC'06), pp. 594–603, 2006. `arXiv:quant-ph/0511013v1`

[12] Masahito Hayashi, Kazuo Iwama, Harumichi Nishimura, Rudy Raymond, Shigeru Yamashita, "Quantum Network Coding," Proceedings of the 24th International Symposium on Theoretical Aspects of Computer Science (STACS'07), pp. 610–621, 2007. `arXiv:quant-ph/0601088v2`

[13] Iordanis Kerenidis, "Quantum Encodings and Applications to Locally Decodable Codes and Communication Complexity," PhD thesis, University of California at Berkeley, 2004.

[14] Iordanis Kerenidis, Ronald de Wolf, "Exponential Lower Bound for 2-Query Locally Decodable Codes via a Quantum Argument," *J. Comput. Syst. Sci.*, vol. 69, 3, pp. 395–420, 2004. `arXiv:quant-ph/0208062v2`

[15] Stephanie Wehner, Ronald de Wolf, "Improved Lower Bounds for Locally Decodable Codes and Private Information Retrieval," Automata, Languages and Programming, pp. 1424–1436, 2005. `arXiv:quant-ph/0403140v2`

[16] Avraham Ben-Aroya, Oded Regev, Ronald de Wolf, "A Hypercontractive Inequality for Matrix-Valued Functions with Applications to Quantum Computing and LDCs," Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS'08), pp. 477–486, 2008. `arXiv:0705.3806v2`

[17] Scott Aaronson, "The Learnability of Quantum States", *Proc. Roy. Soc. London Ser. A*, vol. 463, no. 2088, pp. 3089–3114, 2007. `arXiv:quant-ph/0608142v3`

[18] Andrew Chi-Chin Yao, "Probabilistic computations: Toward a unified measure of complexity," Proceedings of the 18th Annual IEEE Symposium on Foundations of Computer Science (SFCS'77), pp. 222–227, 1977.

[19] Giuliano Benenti, Giulio Casati, Giuliano Strini, "Principles of Quantum Computation and Information," vol. 1, World Scientific, 2004.

[20] Eric W. Weisstein, "Stirling's Approximation," MathWorld. `http://mathworld.wolfram.com/StirlingsApproximation.html`

[21] Eric W. Weisstein, "Sphere Point Picking," MathWorld. `http://mathworld.wolfram.com/SpherePointPicking.html`

[22] Subrahmanyan Chandrasekhar, "Stochastic Problems in Physics and Astronomy," *Reviews of Modern Physics*, vol. 15, no. 1, 1943.

[23] Barry D. Hughes, "Random Walks and Random Environments," vol. 1, Clarendon Press, 1995.

[24] Larry C. Grove, Clark T. Benson, "Finite Reflection Groups", 2nd Ed., Springer, 1985.

[25] Walter W. Rouse Ball, H.S.M. Coxeter, "Mathematical Recreations and Essays," 13th Ed., Courier Dover Publications, 1987.

[26] William K. Wootters, Brian D. Fields, "Optimal State-Determination by Mutually Unbiased Measurements," *Annals of Physics*, vol. 191, Issue 2, pp. 363–381, 1989.

[27] Joseph M. Renes, Robin Blume-Kohout, Andrew J. Scott, Carlton M. Caves, "Symmetric Informationally Complete Quantum Measurements," *J. Math. Phys.*, vol. 45, pp. 2171–2180, 2004. `arXiv:quant-ph/0310075v1`

[28] Gen Kimura, "The Bloch vector for N-level systems," *Physics Letters A*, vol. 314, Issues 5–6, pp. 339–349, 2003. `arXiv:quant-ph/0301152v2`

[29] Gen Kimura, Andrzej Kossakowski, "The Bloch-Vector Space for N-Level Systems: the Spherical-Coordinate Point of View," *Open Syst. Inf. Dyn.*, vol. 12, no. 3, pp. 207–229, 2005. `arXiv:quant-ph/0408014v2`

[30] Mark S. Byrd, Navin Khaneja, "Characterization of the Positivity of the Density Matrix in Terms of the Coherence Vector Representation," *Phys. Rev. A*, vol. 68, Issue 6, 062322, 2003. `arXiv:quant-ph/0302024v2`

[31] Foek T. Hioe, Joseph H. Eberly, "N-Level Coherence Vector and Higher Conservation Laws in Quantum Optics and Quantum Mechanics," *Phys. Rev. Lett.*, vol. 47, pp. 838–841, 1981.

[32] Ingemar Bengtsson, Åsa Ericsson, "Mutually Unbiased Bases and the Complementarity Polytope," *Open Syst. Inf. Dyn.*, vol. 12, no. 2, pp. 107–120, 2005. `arXiv:quant-ph/0410120v1`

[33] Michael A. Nielsen, Isaac L. Chuang, "Quantum Computation and Quantum Information," Cambridge University Press, 2000.

[34] Asher Peres, "Quantum Theory: Concepts and Methods," Kluwer Academic Publishers, 2002.

[35] Roger A. Horn, Charles R. Johnson, "Matrix Analysis," Cambridge University Press, 1985.