Classical Verification of Quantum Computations

Manideep Mamindlapally

August 2022

Abstract

These are notes on a couple of recent works [MF16, Mah18] related to the QPIP$_k$ interactive proof complexity class. We pursue a step-by-step breakdown of the proofs, along with some illustrations for easy intuition.

1 Introduction

An interactive proof system is a setting between two parties. A computationally inferior *verifier* likes to solve a language L, by taking help from a computationally superior *prover*. The prover can however be malicious and can't be trusted. So he will have to compute a solution and also convince the verifier of the solution through a series of message exchanges. Different such interactive systems have been designed and studied for different levels of computational abilities with the prover and the verifier. Some of them are IP where the verifier is capable of polynomial time classical computations and the prover is capable of any classical computation; QIP where the verifier can do ploynomial time quantum computations and the prover can do any quantum computation. The sets of languages that can be solved by them are usually represented by the same letters **IP** and **QIP** respectively. It is worth while to study the computational span of these classes in terms of conventional classes defined in terms of space and time resources. It turns out **IP** and **QIP** are actually both equally as strong as **PSPACE**. In this report we study the QPIP$_k$ interactive proof system and investigate its power in terms of space and time.

1.1 QPIP$_k$ definition

In the QPIP interactive setting, the prover who is capable of performing polynomial time quantum computations, tries to convince a verifier who is capable of performing polynomial time classical computations. In addition, the verifier has combined access (with the prover) to a *k qubit* quantum system over which he can perform quantum operations - unitary transforms and measurements. Both of them are allowed to exchange only a polynomial number of classical messages. The set of problems that can be solved by the verifier is the complexity class $\mathbf{QPIP}_k$.

In this work we talk about two key results that aim to capture the computational power of this QPIP framework. The first of which is

$$\mathbf{QPIP}_1 = \mathbf{BQP} \tag{1}$$

The proof of which consists of two parts. The first part $\mathbf{QPIP}_1 \subseteq \mathbf{BQP}$ is trivial. In fact, $\mathbf{QPIP}_k \subseteq \mathbf{BQP}$ for all integers k, because if there is an a language L where the prover is able to convince the verifier into believing, he should himself be able to solve the language. The second part of the proof $\mathbf{BQP} \subseteq \mathbf{QPIP}_1$ is non trivial. It was proved in [MF16, MNS16]. The proof uses the $\mathbf{QMA}-$completeness of the k LOCAL HAMILTONIAN problem [KSVV02], by extension the 2 LOCAL HAMILTONIAN problem [KKR06] and the 2 LOCAL ZX HAMILTONIAN problems [BL08]. The idea here is to convert an instance $x \in L$ of a $\mathbf{BQP}$ problem to a 2 LOCAL ZX HAMILTONIAN instance H_x, since $\mathbf{BQP} \subseteq \mathbf{QMA}$. The prover ($\mathbf{BQP}$ capable) determines the ground state of H_x. Due to the 2-local nature of the hamiltonian, the verifier only needs to measure qubits locally in Z or X bases. He can do this by fetching each qubit from the prover, one at a time. We will discuss this in more detail in Section ??.

We build upon this result in the Section ?? by showing that

$$\mathbf{QPIP}_0 = \mathbf{BQP} \tag{2}$$

under the assumption of the existence of certain *trapdoor claw free function* families. Here too, the first part $\mathbf{QPIP}_0 \subseteq \mathbf{BQP}$ is trivial as already seen above. For the second part, $\mathbf{BQP} \subseteq \mathbf{QPIP}_0$, [Mah18] emulate a similar form of reduction as in [MF16] but outsource the quantum measurement step completely from the verifier to the prover. They develop a framework that exploits the properties of *trapdoor claw free function* families and their *injective invariance*, to build a MEASUREMENT PROTOCOL that ensures that the prover correctly performs quantum measurement in the basis states desired by the verifier. Unfortunately, though, no one has been able to construct a *trapdoor claw free function* family so far. They present a weaker family, the *extended trapdoor claw free function* family that can be and has been realised, but some of their properties rely on the assumption that a $\mathbf{BQP}$ machine cannot solve the *learning with errors (LWE)* problem. We break down the proofs completely and supplement them with an illustration for an easy intuition in Section ??.

2 Result: $\mathbf{BQP} \subseteq \mathbf{QPIP}_1$ [MF16, MNS16]

Towards proving this result, we use a $\mathbf{QMA}$-complete problem, the k LOCAL HAMILTONIAN PROBLEM. A *k local Hamiltonian* is an operator that can be expressed as a sum of non-negative Hermitian operators, that all act on atmost k qubits. A 2 local ZX Hamiltonian is one where the constituent Hermitians are only made of Pauli operators Z or X and act on atmost two of the qubits. The 2 LOCAL ZX HAMILTONIAN PROBLEM is a membership problem of a language of these 2 local ZX Hamiltonians with low ground energy.

Definition 1 (2 LOCAL ZX HAMILTONIAN PROBLEM [BL08]) *It is the membership problem in language L of a hamiltonian H_{ZX} of the form*

$$H_{ZX} = \sum_i h_i Z_i + \sum_i \delta_i X_i + \sum_{i<j} J_{ij} Z_i X_j + \sum_{i<j} K_{ij} X_i Z_j \tag{3}$$

with h_i, δ_i, J_ij, $K_ij \in \mathbb{R}$

$$H_{ZX} \in L \qquad \iff H \text{ has an eigenvalue not exceeding } a \tag{4}$$

$$H_{ZX} \notin L \qquad \iff H \text{ has no eigenvalue exceeding } b \tag{5}$$

where $0 \leq a < b$ and $b - a = \Omega(n^{-\alpha})$ and $\alpha > 0$ is a constant.

[KSVV02] showed that the 5 LOCAL HAMILTONIAN PROBLEM is **QMA**-complete. In a later iteration [KKR06] showed **QMA**-completeness of the 2 LOCAL HAMILTONIAN PROBLEM. The result was further extended in [BL08] to **QMA**-completeness of the 2 LOCAL ZX HAMILTONIAN PROBLEM. We employ the reduction $\mathcal{R}$ from [BL08], that converts a QMA instance x, and its certificate $|\xi_x\rangle$ to hamiltonian H_x and its eigenstate (with the smallest eigen value) $|\eta_x\rangle$.

Remember that our goal was to prove $\mathbf{BQP} \subseteq \mathbf{QPIP}_1$. We take a language L contained in $\mathbf{BQP}$, and try to show that $L \in \mathbf{QPIP}_1$. Towards that we use the fact that L is also in $\mathbf{QMA}$ (since $\mathbf{BQP} \subseteq \mathbf{QMA}$), with a trivial verification certificate $|\xi\rangle = |0\rangle$ for any input instance x, and the corresponding verification circuit V_x. We can use $\mathcal{R}$ to reduce V_x, $|\xi\rangle = |0\rangle$ in polynomial time to the 2 LOCAL ZX HAMILTONIAN instance H_x, with the eigenstate $|\eta_0\rangle$. In the QPIP interactive proof setting, the BQP capable prover is thus able to establish H_x as well as $|\eta_0\rangle$, while the verifier only computes H_x because he can do only classical computation. Additionally, the prover is already able to determine whether or not $x \in L$ by passing $|0\rangle$ through the circuit V_x and measuring the output.

Now, see that the Hamiltonian H_x, from definition 1 comprises of a number of hermitians, that are all Pauli operators Z,X acting on atmost two qubits. If we represent H_x as sum of such Pauli terms S with coefficients d_s,

$$H_x = \sum_{S \text{ is of the form in definition 1}} d_S S \tag{6}$$

Here $d_S \in \mathbb{R}$ since all the coefficients in definition 1 are real. Let us define

$$\begin{aligned} H'_x &:= H_x + |d_S| I & (7)\\ &= \sum_S |d_S|(I + sign(d_S)S) & (8)\\ &= \sum_S 2|d_S| P_S & (9) \end{aligned}$$

where P_S is a projection operator $P_S := \frac{I+sign(d_S)S}{2}$. Now defining another operator which is a normalised version of H'_x

$$\begin{aligned} H''_x &:= \frac{1}{2\sum_S |d_S|} H'_x & (10)\\ &= \sum_S \pi_S P_S & (11) \end{aligned}$$

where $\pi_S = \frac{|d_S|}{\sum_S |d_S|}$ is a probability term.

PROTOCOL $\mathscr{P}_1$ (for input instance "$x \in L$?")

1. The prover and verifier both determine (can do in classical polynomial time) the description of the quantum circuit V_x of $\mathbf{BQP}$ language L.

2. The verifier uses $\mathcal{R}$ to convert circuit description V_x to a 2 local ZX Hamiltonian H_x.
The prover uses $\mathcal{R}$ to convert V_x to H_x and also convert the trivial certificate state $|0\rangle$ to eigenstate $|\eta_0\rangle$.

$|\eta_0\rangle$ has eigenvalue greater than b or less than a depending on if $x \notin L$ or $x \in L$ respectively. Thus, the verifier is to evaluate $\langle\eta_0|H_x|\eta_0\rangle$. This can be equivalently expressed in terms of the acceptance probability of projection operator H''_x. See eq. (16)

3. The verifier chooses a Pauli operator S with probability π_S from the 2-local expansion of H_x.

4. The prover sends over each of the n qubits of $|\eta_0\rangle$ one by one.

5. For each i^{th} qubit of $|\eta_0\rangle$ received, if the Pauli S contains Z or X on that location i, the verifier measures the i^{th} qubit in the corresonding Z or X basis eigen states.

6. There are atmost two such locations i. If the product of the results equals $-sign(d_S)$, the verifier returns ACCEPT. Else REJECT.

According to this procedure $\mathscr{P}_1$, the probability of acceptance p_{acc} is

$$p_{acc} = 1 - \langle\eta_0|H''_x|\eta_0\rangle \tag{12}$$

$$= 1 - \frac{1}{2\sum_S |d_S|}\langle\eta_0|H'_x|\eta_0\rangle \tag{13}$$

$$= 1 - \frac{1}{2\sum_S |d_S|}\langle\eta_0|(H_x + \sum_S |d_S|I)|\eta_0\rangle \tag{14}$$

$$= 1 - \frac{1}{2\sum_S |d_S|}(\langle\eta_0|H_x|\eta_0\rangle + \sum_S |d_S|) \tag{15}$$

$$= \frac{1}{2} - \frac{\langle\eta_0|H_x|\eta_0\rangle}{2\sum_S |d_S|} \tag{16}$$

if $x \in L$, from definition 1

$$p_{acc} \geq \frac{1}{2} - \frac{a}{2\sum_S |d_S|} \tag{17}$$

if $x \notin L$

$$p_{acc} \leq \frac{1}{2} - \frac{b}{2\sum_S |d_S|} \tag{18}$$

Where $b - a \geq \frac{1}{\text{poly}(|x|)}$. So, by repeating the procedure above polynomial number of times, $x \in L$ and $x \notin L$ can be separated.

PROTOCOL $\mathscr{P}_2$ (for input instance "$x \in L$?")

1. The prover and verifier both determine (can do in classical polynomial time) the description of the quantum circuit V_x of **BQP** language L.

2. Perform steps 2-6 of PROTOCOL $\mathscr{P}_1$ p ($p = \text{poly}(|x|)$) times and count the number of ACCEPTs n_a.

3. If $n_a \geq (\frac{1}{2} - \frac{a+b}{4\sum_S |d_S|})p$, the verificer ACCEPTs. Else REJECTs.

Protocol $\mathscr{P}_2$ amplifies the success probability of $\mathscr{P}_1$ to an error exponentially small in $|x|$. Observe that throughout $\mathscr{P}$ and $\mathscr{P}_2$, the verifier has only one qubit available to him at a time, over which he performs measurements only in Z or X basis. The outcome is that we were able to solve a **BQP** language L entirely using a QPIP$_1$ framework $\mathscr{P}_2$. Thus proving $\mathbf{BQP} \subseteq \mathbf{QPIP}_1$.

3 Result II [Mah18]

See that in protocol $\mathscr{P}_1$ in the previous section, the verifier makes use of exactly one qubit register over which he performs measurements in basis eigen states of Z or X. Consequently we were able to show that any **BQP** language L is in $\mathbf{QPIP}_1$. In this part of the report we will see if L is in $\mathbf{QPIP}_0$. For that we need a framework to outsource the poscession and measurement of the single qubit register at the verifier to the prover who is already **BQP** capable. Unfortunately though, at this point of time we don't have any such proof. However, we will look at [Mah18]'s version of MEASUREMENT PROTOCOL that offers a mechanism to reliably outsource the measurement of a state in a desired basis. It makes use of *trapdoor claw free function families*. Unfortunately though, we do not know any constructions of this family. In [Mah18], they use an approximate version, the *extended trapdoor claw free function family*, that can be efficiently constructed and used but rely on an assumption that the "Learning with errors" problem cannot be solved by a **BQP** machine. This is a commonly used assumption several other cryptographic protocols too.

The key idea with the MEASUREMENT PROTOCOL is to ensure that for any behaviour of the Prover $\mathbb{P}$, there is a valid quantum state ρ, whose measurement statistics $\mathcal{D}_{\rho,h}$ over the bases desired by the verifier h, match the actual statistics obtained $\mathcal{D}_{\mathbb{P},h}$

$$\mathcal{D}_{\rho,h} = \mathcal{D}_{\mathbb{P},h} \tag{19}$$

Let's start by revising the definitions of some preliminaries of [Mah18].

3.1 Preliminaries

3.1.1 Trapdoor Claw Free Families

From the definition in [Mah18], trapdoor claw free families are those of the form

$$\mathcal{F} = \{f_{k,b} : \mathcal{X} \to \mathcal{Y}\}_{k,b} \tag{20}$$

and

- Each of $f_{k,0}$ and $f_{k,1}$ are **injective** and their images are equal i.e., the functions are **two-to-one**.
- There is a classical polynomial time **generation** procedure $GEN_{\mathcal{F}}$, generates a key k and a trapdoor t_k.
$$(k, t_k) \leftarrow GEN_{\mathcal{F}} \tag{21}$$
- There is a quantum polynomial time **peperation** procedure $SAMP_{\mathcal{F}}$ that produces $f_{k,b}(x)$ for given inputs k, b and x.
$$|b\rangle|x\rangle|0\rangle|k\rangle \xrightarrow{SAMP_{\mathcal{F}}} |b\rangle|x\rangle|f_{k,b}(x)\rangle|k\rangle \tag{22}$$

A claw (x_0, x_1) is defined as one if there exists an output y for which $f_{k,0}(x_0) = f_{k,1}(x_1)$.

- Given an output y and function key k, it is **claw-free** i.e., it is computationally hard to find the claw (x_0, x_1) in quantum polynomial time.
- Only using the trapdoor t_k, can one invert. There exists an **inverting** function $INV_{\mathcal{F}}$ which using the trapdoor t_k , output y, inverts back to the input claw (x_0, x_1).

$$INV_{\mathcal{F}}(t_k, y) = (x_0, x_1) \tag{23}$$

- $\mathcal{F}$ satisfies the **hard core bit property**. It is computationally hard for a quantum computer to find a claw (x_0, x_1) and $d \in \{0,1\}^w$ in polynomial time such that $d \cdot (x_0 \oplus x_1) = 0$ with non-negligible advantage over $\frac{1}{2}$.

3.1.2 Trapdoor Injective Function Families

From the definitions in [Mah18], trapdoor injective function families are those of the form

$$\mathcal{G} = \{g_{k,b} : \mathcal{X} \rightarrow \mathcal{Y}\}_{k,b} \tag{24}$$

and

- Each g_k, b is **injective**, $g_{k,0}$ and $g_{k,1}$ have disjoint ranges i.e., they are **one-to-one**.
- There is a classical polynomial time **generation** procedure $GEN_{\mathcal{G}}$, generates a key k and a trapdoor t_k.

$$(k, t_k) \leftarrow GEN_{\mathcal{G}} \tag{25}$$

- There is a quantum polynomial time **peperation** procedure $SAMP_{\mathcal{G}}$ that produces $g_{k,b}(x)$ for given inputs k, b and x.

$$|b\rangle|x\rangle|0\rangle|k\rangle \overset{SAMP_{\mathcal{G}}}{\longrightarrow} |b\rangle|x\rangle|g_{k,b}(x)\rangle|k\rangle \tag{26}$$

- Given an output y and function key k, it is computationally hard to find the preimages b, x_b in quantum polynomial time.
- Only using the trapdoor t_k, can one invert. There exists an **inverting** function $INV_{\mathcal{G}}$ which using the trapdoor t_k , output y, inverts back to the preimage x_b and bit b.

$$INV_{\mathcal{G}}(t_k, y) = (b, x_b) \tag{27}$$

3.1.3 Injective Invariance

A trapdoor claw free family $\mathcal{F}$ is said to be injective invariant with a trapdoor injective family $\mathcal{G}$ if for a given function key k, it is computationally hard for a **BQP** machine to determine if k was sampled from which of the families, unless you know the trapdoor t_k.

3.2 Some Relaxed assumptions

As pointed out earlier, we unfortunately do not know of any constructions of the trapdoor injective functions that satisfy the requirements posed here. In [Mah18], they use these primitives to build a MEASUREMENT PROTOCOL (c.f. Sec ??). They also go on to use a relaxed version, the *extended claw free function families* and *extended trapdoor injective function families* that can be constructed on a more pragmatic assumption that the LEARNING WITH ERRORS problem is difficult to solve for a quantum computer in polynomial time. The latter assumption is more reasonable, widely believed to be true and is also used in several other security protocols as well. For the initial part of the report, for mathematical simplicity purposes, we will stick to the former assumption about the existence of such functions. This allows for an easier intuitive understanding. More precisely our assumptions are

1. There exists a trapdoor claw free function family $\mathcal{F}$ satisfying the definitions of sec ??

2. There exists a trapdoor injective function family $\mathcal{G}$ such that $\mathcal{F}$ and $\mathcal{G}$ are injective invariant.

3.3 The Measurement Protocol

In this section we look at the MEASUREMENT PROTOCOL used in [Mah18] and perform an illustration for a single qubit measurement. Remember that we are trying to emulate step 5 of protocol $\mathscr{P}_1$. The verifier picks a Pauli S and has the corresponding bases of measurement decided for each of the qubit. It is only at most two qubits, that the verifier would want to measure. For simplicity, we let the verifier measure in the standard Z bases over the qubits not mentioned. The prover meanwhile doesn't know which Pauli the verifier picked, but knows the Hamiltonian H_x and the eigenstate $|\eta_0\rangle$.

MEASUREMENT PROTOCOL $\mathscr{M}$

Initially,

1. The verifier $\mathbb{V}$ decided the bases of measurement $h \in \{0,1\}^n$. $h_i = 1$ if X bases are needed for measurement and $h_i = 0$ otherwise.

2. The prover $\mathbb{P}$ has the state ρ which is a combination of three key registers. The first is an n qubit *committed bit* register, initiated to $|\eta_0\rangle\langle\eta_0|$. The second register is what we call a *preimage* register initialised to a fully superposed state of all xs. The third is the the *commitment string* register used to later store the function outputs ys, also initialised to to all $|0\rangle$ state. Each register has provision for ancillary qubits too. Let's say there is a fourth register to later store the function key k as well, also initialised to zero.

$$\rho \leftarrow \frac{1}{|\mathcal{X}^n|} \sum_{x^n \in \mathcal{X}^n} |\eta_0\rangle |x^n\rangle |0^e\rangle |0^e\rangle \tag{28}$$

The protocol itself proceeds in the following steps

(1) For $i = 1$ to n, $\mathbb{V}$ generates function keys k_i and trapdoor t_{k_i} from $\mathcal{G}$ of $\mathcal{F}$ depending on the desired bases of measurement h_i.

$$(k_i, t_{k_i}) \leftarrow \begin{cases} GEN_{\mathcal{G}} & \text{if } h_i = 0 \\ GEN_{\mathcal{F}} & \text{if } h_i = 1 \end{cases} \tag{29}$$

(2) $\mathbb{V}$ sends to $\mathbb{P}$ the key vector k of all keys k_i.

(3) $\mathbb{P}$ sets the fourth register to key vector k and passes ρ through the $SAMP$ state preparation procedure.

(4) $\mathbb{P}$ measures the the third register a.k.a. *commitment string* register in the Z bases to get commitment string y_i for each $i = 1$ to n.

(5) $\mathbb{P}$ sends over each y_i to $\mathbb{V}$

(6) $\mathbb{V}$ computes the inversions of all y_i using the trapdoors t_{k_i}.
If $h_i = 0$, $\mathbb{V}$ computes $(b_i, x_{b_i}) = INV_{\mathcal{G}}(y_i, t_{k_i}$
If $h_i = 1$, $\mathbb{P}$ computes $(x_{0,y_i}, x_{1,y_i}) = INV_{\mathcal{F}}(y_i, t_{k_i})$

(7) $\mathbb{V}$ decides to perform either a TEST ROUND or a MEASUREMENT ROUND with half probability each.

If it's the test round,
TEST ROUND

(T1) $\mathbb{P}$ measures the first and second registers in the standard Z bases to get outputs b'_i, x'_i for each i.

(T2) $\mathbb{P}$ sends each b'_i, x'_i to $\mathbb{V}$

(T3) For each i,
If $h_i = 0$ $\mathbb{V}$ tests if $(b_i, x_{b_i}) == (b'_i, x'_{b_i})$?
If $h_i = 1$ $\mathbb{V}$ tests if $(0, x_{0,y_i}) == (b'_i, x'_{b_i})$ (or) $(1, x_{1,y_i}) == (b'_i, x'_{b_i})$?.
$\mathbb{V}$ aborts the protocol if either of the test fails.

MEASUREMENT ROUND

(M1) $\mathbb{P}$ measures the first and second registers in the hadamard X bases to get outputs $\overline{b}'_i$, d'_i for each i.

(M2) $\mathbb{P}$ sends each $\overline{b}'_i$, x'_i to $\mathbb{V}$

(M3) For each i,
If $h_i = 0$, $\mathbb{V}$ ignores $\overline{b}'_i$, x'_i and calculates $m_i \leftarrow b_i$
If $h_i = 1$, $\mathbb{V}$ calculates $m_i \leftarrow b'_i \oplus d'_i \cdot (x_{0,y_i} \oplus x_{1,y_i})$.
and returns the cumulative result of all m_i as the measurement result.

3.3.1 Illustration

For this illustration purpose let us start with a single qubit state $|\eta_0\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$. Also let us assume that the prover behaves honest to the protocol. We will drop the subscript i in the variables h_i, y_i, b_i we use (in this subsection), since there is only a single qubit.

At the start of $\mathscr{M}$, $\mathbb{V}$ chooses the basis state for measurement. Let us split the analysis for $h = 0$ and $h = 1$ seperately.
Case I: $h = 0$
Initially, $\mathbb{P}$ starts with ρ

$$\rho \leftarrow \frac{1}{\sqrt{|\mathcal{X}|}} \sum_{x \in \mathcal{X}} \sum_{b \in \{0,1\}} \alpha_b |b\rangle |x\rangle |0^e\rangle |0^e\rangle \tag{30}$$

In step (1), $\mathbb{V}$ samples (k, t_k) from $GEN_{\mathcal{G}}$, of which k is sent to $\mathbb{P}$ in step (2). After step (3), the state at $\mathbb{P}$ is

$$\rho \leftarrow SAMP\left(\frac{1}{\sqrt{|\mathcal{X}|}}\sum_{x\in\mathcal{X}}\sum_{b\in\{0,1\}}\alpha_b|b\rangle|x\rangle|0^e\rangle|k\rangle\right) \tag{31}$$

$$= \frac{1}{\sqrt{|\mathcal{X}|}}\sum_{x\in\mathcal{X}}\sum_{b\in\{0,1\}}\alpha_b|b\rangle|x\rangle|g_{k,b}(x)\rangle|k\rangle \tag{32}$$

In step (4), $\mathbb{P}$ measures the third register in Z bases to get output y. At this stage, ρ collapses to

$$\rho \leftarrow \begin{cases} |0\rangle|x_{0,y}\rangle & (y = g_{k,0}(x_{0,y}) \text{ for some } x_{0,y} \in \mathcal{X}) \text{ w.p.}|\alpha_0|^2 \\ |1\rangle|x_{1,y}\rangle & (y = g_{k,1}(x_{1,y}) \text{ for some } x_{1,y} \in \mathcal{X}) \text{ w.p.}|\alpha_1|^2 \end{cases} \tag{33}$$

[1] Here $x_{b,y}$ is such that $g_{k,b}(x_{b,y}) = y$. This is because from the one-to-one property of $\mathcal{G}$, every y has only one preimage $(b, x_{b,y})$. y could have been measured from $g_{k,b}(\cdot)$ of either of b with $|\alpha_b|^2$ probability because of the superposition in previous step.
The measured y is sent over to the $\mathbb{V}$ in step (5), who in step (6) performs the inversion operation $INV_{\mathcal{G}}(y, t_k)$ to get the function inputs $(b, x_{b,y})$. Either of $b = 0$ or 1 could have been the result with probabilities $|\alpha_0|^2$ and $|\alpha_1|^2$ respectively. Nonetheless, these values match the state $|b\rangle|x_{b,y}\rangle$ (in ρ) that $\mathbb{P}$ has.

If it is the Test Round, in step (T1) $\mathbb{P}$ measures the first and second registers of ρ in Z bases (in (33)) to get $b' = b$ and $x' = x_{b,y}$ always. On sending these values to the $\mathbb{V}$ in step (T2), he performs a test in step (T3) which passes because of the above reason.

If it is the Measurement Round, in step (M1), $\mathbb{P}$ measures the first and second registers of ρ in X bases(in (33)). These results should not matter, as $\mathbb{V}$ directly calculates the measurement result $m = b$ obtained in step (6). As already seen, each of the b is output with $|\alpha_b|^2$ probability each. Therefore $\mathcal{D}_{\mathbb{P},h=0} = \{|\alpha_0|^2, |\alpha_1|^2\}$. This equals the statistics of measurement if ρ were directly measured in the Z basis, $\mathcal{D}_{\rho,h=0} = \{|\alpha_0|^2, |\alpha_1|^2\}$. Therefore $\mathcal{D}_{\mathbb{P},h=0} = \mathcal{D}_{\rho,h=0}$.
Case II: $h = 1$
Initially, $\mathbb{P}$ starts with ρ

$$\rho \leftarrow \frac{1}{\sqrt{|\mathcal{X}|}}\sum_{x\in\mathcal{X}}\sum_{b\in\{0,1\}}\alpha_b|b\rangle|x\rangle|0^e\rangle|0^e\rangle \tag{34}$$

In step (1), $\mathbb{V}$ samples (k, t_k) from $GEN_{\mathcal{F}}$, of which k is sent to $\mathbb{P}$ in step (2). After step (3), the state at $\mathbb{P}$ is

$$\rho \leftarrow SAMP\left(\frac{1}{\sqrt{|\mathcal{X}|}}\sum_{x\in\mathcal{X}}\sum_{b\in\{0,1\}}\alpha_b|b\rangle|x\rangle|0^e\rangle|k\rangle\right) \tag{35}$$

$$= \frac{1}{\sqrt{|\mathcal{X}|}}\sum_{x\in\mathcal{X}}\sum_{b\in\{0,1\}}\alpha_b|b\rangle|x\rangle|f_{k,b}(x)\rangle|k\rangle \tag{36}$$

In step (4), $\mathbb{P}$ measures the third register in Z bases to get output y. At this stage, ρ collapses to

$$\rho \leftarrow \sum_{b\in\{0,1\}}\alpha_b|b\rangle|x_{b,y}\rangle \tag{37}$$

[1]For simplicity we dropped the third and fourth registers that have already collapsed to $|y\rangle$ and $|k\rangle$ respectively.

[2] Here $x_{0,y}$ and $x_{1,y}$ is a claw such that $f_{k,0}(x_{0,y}) = f_{k,1}(x_{1,y}) = y$. This is because from the two-to-one property of $\mathcal{F}$, every y has two preimages that form a claw.
The measured y is sent over to the $\mathbb{V}$ in step (5), who in step (6) performs the inversion operation $INV_{\mathcal{F}}(y, t_k)$ to get the input claw $(x_{0,y}, x_{1,y})$.

If it is the TEST ROUND, in step (T1) $\mathbb{P}$ measures the first and second registers of ρ in Z bases (in (37)) to get either $(b' = 0, x' = x_{0,y})$ or $(b' = 1, x' = x_{1,y})$ with probabilities $|\alpha_0|^2$ and $|\alpha_1|^2$ respectively (because of the superposition in (37)). On sending these values to $\mathbb{V}$ in step (T2), he performs a test in step (T3) which passes because of the above reason.

If it is the MEASUREMENT ROUND, in step (M1), $\mathbb{P}$ measures the first and second registers of ρ in X bases(in (33)). This is equivalent to first performing a hadamard gate operation and then measuring in the Z bases. Performing the Hadamard operation to ρ in (37) results in

$$H \otimes H \left(\sum_{b \in \{0,1\}} \alpha_b |b\rangle |x_{b,y}\rangle \right) \tag{38}$$

$$= (H \otimes I)(I \otimes H) \left(\sum_{b \in \{0,1\}} \alpha_b |b\rangle \otimes X^{x_{b,y}} |0\rangle \right) \tag{39}$$

$$= (H \otimes I) \left(\sum_{b \in \{0,1\}} \alpha_b |b\rangle \otimes H X^{x_{b,y}} |0\rangle \right) \tag{40}$$

$$= (H \otimes I) \left(\sum_{b \in \{0,1\}} \alpha_b |b\rangle \otimes Z^{x_{b,y}} H |0\rangle \right) \tag{41}$$

$$= (H \otimes I) \left(\sum_{b \in \{0,1\}} \alpha_b |b\rangle \otimes Z^{x_{b,y}} \left(\sum_{d \in \mathcal{X}} \frac{1}{\sqrt{|\mathcal{X}|}} |d\rangle \right) \right) \tag{42}$$

$$= \frac{1}{\sqrt{|\mathcal{X}|}} (H \otimes I) \left(\sum_{d \in \mathcal{X}} \sum_{b \in \{0,1\}} \alpha_b |b\rangle \otimes Z^{x_{b,y}} |d\rangle \right) \tag{43}$$

$$= \frac{1}{\sqrt{|\mathcal{X}|}} (H \otimes I) \left(\sum_{d \in \mathcal{X}} \sum_{b \in \{0,1\}} \alpha_b |b\rangle \otimes (-1)^{d \cdot x_{b,y}} |d\rangle \right) \tag{44}$$

$$= \frac{1}{\sqrt{|\mathcal{X}|}} (H \otimes I) \left(\sum_{d \in \mathcal{X}} \sum_{b \in \{0,1\}} (-1)^{d \cdot x_{b,y}} \alpha_b |b\rangle |d\rangle \right) \tag{45}$$

$$= \frac{1}{\sqrt{|\mathcal{X}|}} (H \otimes I) \left(\sum_{d \in \mathcal{X}} (-1)^{d \cdot x_{0,y}} \alpha_0 |0\rangle |d\rangle + (-1)^{d \cdot x_{1,y}} \alpha_1 |1\rangle |d\rangle \right) \tag{46}$$

$$= \frac{1}{\sqrt{|\mathcal{X}|}} (H \otimes I) \left(\sum_{d \in \mathcal{X}} \left(\alpha_0 |0\rangle + (-1)^{d \cdot (x_{0,y} \oplus x_{1,y})} \alpha_1 |1\rangle \right) \otimes (-1)^{d \cdot x_{0,y}} |d\rangle \right) \tag{47}$$

[2]For simplicity we dropped the third and fourth registers that have already collapsed to $|y\rangle$ and $|k\rangle$ respectively.

$$= \frac{1}{\sqrt{|\mathcal{X}|}}(H \otimes I)\left(\sum_{d\in\mathcal{X}} Z^{d\cdot(x_{0,y}\oplus x_{1,y})}\left(\alpha_0|0\rangle + \alpha_1|1\rangle\right) \otimes Z^{d\cdot x_{0,y}}|d\rangle\right) \tag{48}$$

$$= \frac{1}{\sqrt{|\mathcal{X}|}}(H \otimes I)\left(\sum_{d\in\mathcal{X}}\left(Z^{d\cdot(x_{0,y}\oplus x_{1,y})}\sum_{b\in\{0,1\}}\alpha_b|b\rangle\right) \otimes \left(Z^{d\cdot x_{0,y}}|d\rangle\right)\right) \tag{49}$$

$$= \frac{1}{\sqrt{|\mathcal{X}|}}\sum_{d\in\mathcal{X}}\left(HZ^{d\cdot(x_{0,y}\oplus x_{1,y})}\sum_{b\in\{0,1\}}\alpha_b|b\rangle\right) \otimes \left(Z^{d\cdot x_{0,y}}|d\rangle\right) \tag{50}$$

$$= \frac{1}{\sqrt{|\mathcal{X}|}}\sum_{d\in\mathcal{X}}\left(X^{d\cdot(x_{0,y}\oplus x_{1,y})}H\left(\alpha_0|0\rangle + \alpha_1|1\rangle\right)\right) \otimes \left(Z^{d\cdot x_{0,y}}|d\rangle\right) \tag{51}$$

$$= \frac{1}{\sqrt{|\mathcal{X}|}}\sum_{d\in\mathcal{X}}\left(X^{d\cdot(x_{0,y}\oplus x_{1,y})}\left(\frac{\alpha_0+\alpha_1}{\sqrt{2}}\alpha_0|0\rangle + \frac{\alpha_0-\alpha_1}{\sqrt{2}}\alpha_1|1\rangle\right)\right) \otimes \left(Z^{d\cdot x_{0,y}}|d\rangle\right) \tag{52}$$

$$= \frac{1}{\sqrt{|\mathcal{X}|}}\sum_{d\in\mathcal{X}}X^{d\cdot(x_{0,y}\oplus x_{1,y})}\left(\sum_b \overline{\alpha}_b|b\rangle\right) \otimes Z^{d\cdot x_{0,y}}|d\rangle \tag{53}$$

$$= \frac{1}{\sqrt{|\mathcal{X}|}}\sum_{d\in\mathcal{X}}\left(\sum_b \overline{\alpha}_b X^{d\cdot(x_{0,y}\oplus x_{1,y})}|b\rangle\right) \otimes Z^{d\cdot x_{0,y}}|d\rangle \tag{54}$$

$$= \frac{1}{\sqrt{|\mathcal{X}|}}\sum_{d\in\mathcal{X}}\left(\sum_b \overline{\alpha}_b|b\oplus d\cdot(x_{0,y}\oplus x_{1,y})\rangle\right) \otimes Z^{d\cdot x_{0,y}}|d\rangle \tag{55}$$

where $\overline{\alpha}_b := \frac{\alpha_0+(-1)^b\alpha_1}{\sqrt{2}}$. Now measuring the first and second registers in standard basis gives some d' with probability $\frac{1}{|\mathcal{X}|}$. The state in the first register collapses to

$$\sum_b \overline{\alpha}_b|b\oplus d'\cdot(x_{0,y}\oplus x_{1,y})\rangle \tag{56}$$

Measuring the first qubit in Z basis gives result $\overline{b'} = b\oplus d'\cdot(x_{0,y}\oplus x_{1,y})$ with probability $|\overline{\alpha}_b|^2 = \frac{|\alpha_0+(-1)^b\alpha_1|^2}{2}$. In Step (M2), b' and d' are sent to $\mathbb{V}$ who then cancels the additional $d'\cdot(x_{0,y}\oplus x_{1,y})$ term in b' by XORing with it, thus getting results $m = 0$ or 1 with probabilities $\frac{|\alpha_0+\alpha_1|^2}{2}$ and $\frac{|\alpha_0-\alpha_1|^2}{2}$ respectively.

Therefore $\mathcal{D}_{\mathbb{P},h=1} = \{\frac{|\alpha_0+\alpha_1|^2}{2}, \frac{|\alpha_0-\alpha_1|^2}{2}\}$. This equals the statistics of measurement if ρ were directly measured in the X basis, $\mathcal{D}_{\rho,h=0} = \{\frac{|\alpha_0+\alpha_1|^2}{2}, \frac{|\alpha_0-\alpha_1|^2}{2}\}$. Therefore $\mathcal{D}_{\mathbb{P},h=1} = \mathcal{D}_{\rho,h=1}$. The result of Cases I and II is that $\mathcal{D}_{\mathbb{P},h} = \mathcal{D}_{\rho,h}$.

3.4 General Prover behaviour

In the previous section we have seen an illustration of how the protocol works for a single qubit measurement when the prover behaves honestly i.e., sticks to the protocol. This does provide a good intuition and an easy first step towards understanding the proofs the follow now on. That said, a protocol that works for an honest prover is not of any non trivial value. The entire QPIP interactive setup is based on the verifier not being trustful of the prover. If he was, then he could have relayed any **BQP** language instance to the prover directly and fetched for the answer. In this and the later sections, we will see how THE MEASUREMENT PROTOCOL $\mathscr{M}$ guarantees good *soundness* and *completeness* ratios even for a prover that deviates from the prescribed steps. First, in this section we will analyse the ways in which a prover can cheat and grasp a mathematically sound deviated version of $\mathscr{M}$ that applies to even a cheating Prover.

Honest Prover: In $\mathscr{M}$, the prover initiates the state ρ to (28), and then using the keys k prepares a state using $SAMP$ procedure. Let us represent this operation equivalently as a unitary $U_{C,0}$. An honest prover $\mathbb{P}_0$ would simply apply the operation $U_{C,0}$ on an initial state $|0^n\rangle|0^{nw}\rangle|0^e\rangle|k\rangle$ and then follow steps (4)-(5) and the Test and Measurement rounds as specified by $\mathscr{M}$.

Cheating Prover: Let us look at different attacks or deviations a cheating Prover $\mathbb{P}'$ can induce over $\mathscr{M}$.

1. $\mathbb{P}'$ can perform a unitary U_C over the initial state $|0^n\rangle|0^{nw}\rangle|0^e\rangle|k\rangle$ prior to applying $U_{C,0}$. This unitary matrix provision encompasses all cheating strategies of $\mathbb{P}'$ (like different state initialisation and modifying the key k) upto step (3) i.e., applying $SAMP$ procedure.

2. Applying a unitary $U_{C,0}$ different from what is used by $\mathbb{P}_0$. This covers for $\mathbb{P}'$'s behaviour when he uses a wrong $SAMP$ procedure.
   At this point, $\mathbb{P}'$ measures the third register to get y (step (4)), although he may or may not communicate the exact y over to the $\mathbb{V}$.

3. Applying a unitary U_T at the start of step (T1) of TEST ROUND.
   Here too, $\mathbb{P}'$ can send b_i', x_i' of his choosing to $\mathbb{V}$ in step (T2). This is captured well by a unitary U_T that maps any input state to exactly the desired $|b_i'\rangle|x_i'\rangle$. This requires some additional auxiliary qubits.

4. In case of a Measurement round, applying a unitary U_M at the start of step (M1).
   Here too, $\mathbb{P}'$ can send $\overline{b}_i', d_i'$ of his choosing to $\mathbb{V}$ in step (M2). This is captured well by a unitary U_M that maps any input state to exactly the desired $|\overline{b}_i'\rangle|d_i'\rangle$. This requires some additional auxiliary qubits.

Observe that U_T and U_M only need to act on the first two registers. If $\mathbb{P}'$ wants U_T (or U_M) to be a function of the measured outcome y, this is equivalent to U_T (or U_M) acting on an auxiliary state where the contents of the third register have been copied at the start of the protocol. This construction of U_T and U_M allows them to commute with step (4) i.e., the measurement of the third register to get y. We will use the fact that U_T commutes with measurement of y and interchange the order. This results in a cumulative operation of $U_0 := U_T U_{C,0} U_0$ prior to step (4). To keep it equivalent, we replace the attack U_M in the measurement round by $U := U_T^\dagger U_M$. Any prover $\mathbb{P}'$'s behaviour is therefore characterised by unitaries (U_0, U). The generalised version of protocol $\mathscr{M}$ for general prover behaviours is explored in protocol $\mathscr{M}_1$.

PROTOCOL $\mathscr{M}_1$

(1) For $i = 1$ to n, $\mathbb{V}$ generates function keys k_i and trapdoor t_{k_i} from $\mathcal{G}$ of $\mathcal{F}$ depending on the desired bases of measurement h_i.

$$(k_i, t_{k_i}) \leftarrow \begin{cases} GEN_{\mathcal{G}} & \text{if } h_i = 0 \\ GEN_{\mathcal{F}} & \text{if } h_i = 1 \end{cases} \tag{57}$$

(2) $\mathbb{V}$ sends all keys k to $\mathbb{P}'$, who now prepares an initial state

$$|0^n\rangle|0^{nw}\rangle|0^e\rangle|k\rangle$$

(3) $\mathbb{P}'$ perform unitary operation U_0 which can be simplified as

$$U_0(|0^n\rangle|0^{nw}\rangle|0^e\rangle|k\rangle) = U_{0,k}(|0^n\rangle|0^{nw}\rangle|0^e\rangle) \otimes |k\rangle \tag{58}$$

where $U_{0,k}$ is a unitary that is specific to the key k.

(4) $\mathbb{P}'$ measures the third register in Z basis to obtain y.

(5) $\mathbb{P}'$ sends y to $\mathbb{V}$. $\mathbb{V}$ computes the inversions of all y_i using the trapdoors t_{k_i}.
If $h_i = 0$, $\mathbb{V}$ computes $(b_i, x_{b_i}) = INV_{\mathcal{G}}(y_i, t_{k_i}$
If $h_i = 1$, $\mathbb{P}'$ computes $(x_{0,y_i}, x_{1,y_i}) = INV_{\mathcal{F}}(y_i, t_{k_i})$

(6) $\mathbb{V}$ chooses to perform either TEST ROUND or MEASUREMENT ROUND with half probability each.

If it is the test round,

(T1) $\mathbb{P}'$ measures the first and second registers in the standard Z bases to get outputs b'_i, x'_i for each i. $\mathbb{P}$ sends each b'_i, x'_i to $\mathbb{V}$

(T2) For each i,
If $h_i = 0$ $\mathbb{V}$ tests if $(b_i, x_{b_i}) == (b'_i, x'_{b_i})$?
If $h_i = 1$ $\mathbb{V}$ tests if $(0, x_{0,y_i}) == (b'_i, x'_{b_i})$ (or) $(1, x_{1,y_i}) == (b'_i, x'_{b_i})$?.
$\mathbb{V}$ aborts the protocol if either of the test fails.

If it is the measurment round,

(M1) $\mathbb{P}'$ applies unitary U on the first and second registers.

(M2) $\mathbb{P}'$ measures the first two registers in X basis for each $i = 1$ to n, to get $\overline{b}'_i, d'_i$. $\mathbb{P}'$ sends $\overline{b}'_i, d'_i$ to $\mathbb{V}$ for each i.

(M3) For each i,
If $h_i = 0$, $\mathbb{V}$ calculates $m_i = b_i$ (Irrespective of $\overline{b}'_i$, d'_i)
If $h_i = 0$, $\mathbb{V}$ calculates $m_i = b'_i \oplus d'_i \cdot (x_{0,y_i} + x_{1,y_i})$
$\mathbb{V}$ returns m which is a vector of all m_i.

In case of a test round, step (6) is immediately followed by (T1) and (T2). If the test is to pass, $\mathbb{P}'$'s state at the end of step (4) in the first two registers has to be

$$n_\alpha \left(\bigotimes_{i:h_i=0} |b_i\rangle|x_{b_i,y_i}\rangle \right) \otimes \left(\bigotimes_{i:h_i=1} \left(\sum_{b_i \in \{0,1\}} \alpha_{i,b_i}|b_i\rangle|x_{b_i,y_i}\rangle \right)\right) \tag{59}$$

where α_{i,b_i} could be any complex coefficients and n_α is the corresponding normalising coefficient. Additionally, y that $\mathbb{P}'$ sends to $\mathbb{V}$ has to be same as the one measured y, which is same as the y corresponding to preimages x_{b_i,y_i} in (59). Otherwise, test (T2) fails because, every y_i has exactly one (or two) preimages in the function g_k (or f_k respectively). The entire purpose of the test round is to ensure that the prover's state at the start of measurement round is (59). From now on, we will only concern ourselves with the measurement round and evaluate the statatics of the resulting measurement m, $\mathcal{D}_{\mathbb{P}',h}$. Crucially, we would like to show that this statistic is same as the statistic obatained while measuring some quantum state ρ in h basis $\mathcal{D}_{\rho,h}$.

3.4.1 To Prove

We achieve our goal of showing that $\mathcal{D}_{\mathbb{P}',h} \approx \mathcal{D}_{\rho,h}$ for some ρ via a two part process. Remember that every prover is characterised by just the unitary operators (U_0, U) according to $\mathscr{M}$. In the first part, we show the result for provers $\mathbb{P}'$ characterised by a (U_0, U) for certain X-trivial operators U.

We say that a unitary operation U is X-trivial when U commutes with a measurement operation in Z basis of the first register i.e., *the committed qubit* register. In other words, interchanging the order of operations of U and the Z-basis measurement (of the first register) results in the same quantum state. We will use this fact to prove the following

Claim 2 *For any prover $\mathbb{P}$ characterised by (U_0, U) where U is X-trivial, there exists a quantum state ρ with matching measurement statistics in any given basis h.*

$$\mathcal{D}_{\mathbb{P},h} \approx \mathcal{D}_{\rho,h} \tag{60}$$

Towards arriving at the result, we make use of another claim which will be proved in section ??

Claim 3 *For any prover $\mathbb{P}'$ characterised by (U_0, U), there exists another prover $\mathbb{P}$ characterised by $(U_0, \overline{U})$ where $\overline{U}$ is X-trivial with matching measurement statistics in any given basis h.*

$$\mathcal{D}_{\mathbb{P},h} \approx \mathcal{D}_{\mathbb{P}',h} \tag{61}$$

3.5 Proofs

3.5.1 Proof of Claim 2

Towards this proof, we take the measurement protocol $\mathscr{M}_1$ corresponding to a prover $\mathbb{P}$ characterised by (U_0, U) where U is X-trivial. We design a quantum state ρ whose measurement statistics we show match those obtained in $\mathscr{M}_1$. The X-trivial property of U will be crucial in the techniques employed. First, let us recall the protocol $\mathscr{M}_1$ for prover $\mathbb{P}$. Since we have established that the entire purpose of the TEST ROUND is to collapse $\mathbb{P}$'s state to (59), we will only pursue the MEASUREMENT ROUND here. Moreover, in this section of the proof, we are interested only in the output statistics of the MEASUREMENT ROUND.

PROTOCOL $\mathscr{M}_1$ (*Reduced version from $\mathbb{P}$'s perspective*)

(1) For $i = 1$ to n, $\mathbb{P}$ receives key k_i that is generated from $GEN_{\mathcal{G}}$ if $h_i = 0$, or from $GEN_{\mathcal{F}}$ if $h_i = 1$. ($\mathbb{P}$ *does not know h_i though*)

(2) $\mathbb{P}$ performs U_0 on initial state $|0^n\rangle|0^{nw}\rangle|0^e\rangle|k\rangle$.

(3) $\mathbb{P}$ measures the third register to get output y.
$\mathbb{P}$ sends its value to $\mathbb{V}$.
The Test Round ensures that $\mathbb{P}$'s current state (result of U_0 on the initial state) is of the form in (59).

(4) $\mathbb{P}$ applies U over the first two registers.

(5) $\mathbb{P}$ measures the second register in X basis to get $d_1, d_2, \dots, d_n$.
$\mathbb{P}$ measures the first register in X basis to get $\overline{b}_1, \overline{b}_2, \dots, \overline{b}_n$.

(6) $\mathbb{P}$ sends these values to $\mathbb{V}$.
For each i, if $h_i = 0$,
$\mathbb{V}$ *has meanwhile computed* b_i, x_{b_i,y_i} *by applying* $INV_{\mathcal{G}}$ *on* y *in step 3.*
$\mathbb{V}$ calculates $m_i = b_i$
if $h_i = 1$,
$\mathbb{V}$ *has meanwhile computed the claw* (x_{0,y_i}, x_{1,y_i}) *by applying* $INV_{\mathcal{F}}$ *on* y *in step 3.*
$\mathbb{V}$ calculates $m_i = \bar{b}_i \oplus d_i \cdot (x_{0,y_i} + x_{1,y_i})$
Measurement result $= (m_1, m_2, ..., m_n)$

The statistics of measurement results thus obtained by the verifier who desires to measure in h bases is $\mathcal{D}_{\mathbb{P},h}$. We now design a procedure that generates a state $\rho_h^{(1)}$, whose statistics of measurement in h bases $\mathcal{D}_{\rho_h^{(1)},h}$ matches $\mathcal{D}_{\mathbb{P},h}$. Note that $\rho_h^{(1)}$ is not a quantum state in the traditional sense because the state itself depends on the bases states h. Through a series of small changes to the construction, we aim to construct a state that is well defined and whose measurement statistics match that of $\mathscr{M}_1$. First, the construction for $\rho_h^{(1)}$ is

CONSTRUCTION $\mathscr{C}_1(U_0, U)$

(1) or $i = 1$ to n, generate (k_i, t_{k_i}) from $GEN_{\mathcal{G}}$ if $h_i = 0$, or from $GEN_{\mathcal{F}}$ if $h_i = 1$.

(2) Perform unitary U_0 on an initial state $|0^n\rangle|0^{nw}\rangle|0^e\rangle|k\rangle$.

(3) Measure the third register to get output y.

(4) Apply U over the first and second registers.

(5) Measure the second register in X bases to get d_1, d_2, ... , d_n.

(6) For each $i = 1$ to n,
if $h_i = 0$, do nothing in this step.
if $h_i = 1$,
Get the claw $(x_{0,y_i}, x_{1,y_i}) = INV_{\mathcal{F}}(y_i, t_{k_i})$.
Apply $Z^{d_i \cdot (x_{0,y_i} + x_{1,y_i})}$ to ith qubit in the first register

The resulting state (in the first register) is defined as $\rho_h^{(1)}$.
Measurement result is obtained by measuring $\rho_h^{(1)}$ in the basis specified by h.

To be completed ..

References

[BL08] Jacob D Biamonte and Peter J Love. Realizable hamiltonians for universal adiabatic quantum computers. *Physical Review A*, 78(1):012352, 2008.

[KKR06] Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local hamiltonian problem. *Siam journal on computing*, 35(5):1070–1097, 2006.

[KSVV02] Alexei Yu Kitaev, Alexander Shen, Mikhail N Vyalyi, and Mikhail N Vyalyi. *Classical and quantum computation*. Number 47. American Mathematical Soc., 2002.

[Mah18] Urmila Mahadev. Classical verification of quantum computations. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 259–267. IEEE, 2018.

[MF16] Tomoyuki Morimae and Joseph F Fitzsimons. Post hoc verification with a single prover. *arXiv preprint arXiv:1603.06046*, 2016.

[MNS16] Tomoyuki Morimae, Daniel Nagaj, and Norbert Schuch. Quantum proofs can be verified using only single-qubit measurements. *Physical Review A*, 93(2):022326, 2016.