

In general, g_5 changes $|0^{n+1}1\rangle|x_1x_2\cdots x_n\rangle$ to $|\hat{3}\hat{x}_1\hat{x}_2\cdots \hat{x}_n\rangle$.

(*) To explain our procedure further, for readability, we include more 0s to $|00\rangle|\hat{3}\hat{x}_1\hat{x}_2\hat{x}_3\rangle$, and hereafter we are focused on $|(00)^6|\hat{3}\hat{x}_1\hat{x}_2\hat{x}_3\rangle$.

3) We transform the first two bits 00 in $|(00)^6|\hat{3}\hat{x}_1\hat{x}_2\hat{x}_3\rangle$ to 11 ($= \hat{2}$) and then move them to the end of the qustring, resulting in $|(00)^5|\hat{3}\hat{x}_1\hat{x}_2\hat{x}_3\hat{2}\rangle$. This transformation is carried out as follows. Similarly to $CNOT$, we define $k_1 = NOT \circ SWAP \circ NOT$. Since $k_1(|00\rangle) = |11\rangle$, it suffices to define $f_3 = REMOVE_2 \circ k_1$.

4) In the beginning, our qustring is of the form $|\hat{3}\hat{x}_1\hat{x}_2\hat{x}_3\hat{2}\rangle$ by ignoring the leading bits $(00)^5$. We pay our attention to the qustring located between $\hat{3}$ and $\hat{2}$. We place the last two bits $\hat{2}$ into the location immediately right to $\hat{3}\hat{x}_1$ together with changing $\hat{2}$ to $\hat{3}$. We then obtain $|\hat{3}\hat{x}_1\hat{3}\hat{x}_2\hat{x}_3\rangle$. This process is realized by an appropriate \square_1^{QP} -function in the following fashion.

Let k_2 denote a bijection from $\{0, 1\}^6$ to $\{0, 1\}^6$ satisfying that $k_2(vw\hat{2}) = v\hat{2}w$ if $v, w \in \{\hat{0}, \hat{1}\}$, $k_2(\hat{3}w\hat{2}) = \hat{3}w\hat{3}$ if $w \in \{\hat{0}, \hat{1}\}$, $k_2(w\hat{3}\hat{2}) = w\hat{3}\hat{3}$ if $w \in \{\hat{0}, \hat{1}\}$, and $k_2(vwz) = vwz$ if $v, w, z \neq \hat{2}$. With this k_2 , we define g_5 as

$$g_5(|\phi\rangle) = \begin{cases} |\phi\rangle & \text{if } \ell(|\phi\rangle) \leq 6, \\ \sum_{y:|y|=6} |k_2(y)\rangle\langle y|\phi\rangle & \text{otherwise.} \end{cases}$$

By Lemma 3.7, g_5 belongs to $\widehat{\square_1^{\text{QP}}}$. We then define a quantum function h_5 by setting $h_5 = 2QRec_2[I, g_5, I|h_5, h_5, h_5, h_5]$, namely,

$$h_5(|\phi\rangle) = \begin{cases} |\phi\rangle & \text{if } \ell(|\phi\rangle) \leq 2, \\ \sum_{y \in \{\hat{0}, \hat{1}, \hat{2}, \hat{3}\}} g_5(|y\rangle \otimes h_5(\langle y|\phi\rangle)) & \text{otherwise.} \end{cases}$$

After placing $\hat{3}$ into the right of $\hat{3}\hat{x}_1$, we finally obtain the qustring $|(00)^5|\hat{3}\hat{x}_1\hat{3}\hat{x}_2\hat{x}_3\rangle$.

5) Let us define $h'_5 = h_5 \circ f_3$, which is the compositions of Steps 3)–4). By applying h'_5 repeatedly, we can generally change $|(00)^{n-1}|\hat{3}\hat{x}_1\hat{x}_2\cdots\hat{x}_n\rangle$ to $|\hat{3}\hat{x}_1\hat{x}_2\cdots\hat{3}\hat{x}_n\rangle$. Furthermore, if we apply h'_5 twice to $|(00)^3|\hat{3}\hat{x}_1\hat{3}\hat{x}_2\hat{3}\hat{x}_3\rangle$ in our example, then it is possible to append $|\hat{3}\hat{b}\rangle$ to the end of the qustring by consuming $(00)^3$, and then we obtain $|\hat{3}\hat{x}_1\hat{3}\hat{x}_2\hat{3}\hat{x}_3\rangle|\hat{3}\hat{b}\rangle$. Using $|0^{2p(n)-n+1}1\rangle$ in $|\phi^p\rangle$, we repeat Steps 3)–4) $2p(n) - n + 1$ times to encode the content of the first $p(n) + 1$ tape cells indexed by nonnegative numbers. Returning to our example, if we take $|0^41\rangle$, then this process transforms $|0^41\rangle|(00)^6|\hat{3}\hat{x}_1\hat{x}_2\hat{x}_3\rangle$ into $|0^41\rangle|(00)^2|\hat{3}\hat{x}_1\hat{3}\hat{x}_2\hat{3}\hat{x}_3\rangle|\hat{3}\hat{b}\rangle$. To realize this transform by an appropriate \square_1^{QP} -function, since Steps 3)–4) exclude $|0^41\rangle$, we first need to extend h'_5 to $H_1 = Skip[h'_5]$ by Lemma 3.9. The repetition of Steps 3)–4) is done by the following quantum function f_5 :

$$f_5(|\phi\rangle) = \begin{cases} |\phi\rangle & \text{if } \ell(|\phi\rangle) \leq 1, \\ H_1(|0\rangle \otimes f_5(\langle 0|\phi\rangle) + |1\rangle\langle 1|\phi\rangle) & \text{otherwise.} \end{cases}$$

More generally, f_5 changes $|0^{2p(n)-n+1}1\rangle|(00)^{2p(n)-n+1}|\hat{3}\hat{x}_1\hat{x}_2\cdots\hat{x}_n\rangle$ to $|0^{2p(n)-n+1}1\rangle|\hat{3}\hat{x}_1\hat{3}\hat{x}_2\cdots\hat{3}\hat{x}_n\rangle|\hat{3}\hat{b}\hat{3}\hat{b}\cdots\hat{3}\hat{b}\rangle$ with $p(n) - n + 1$ copies of $\hat{3}\hat{b}\rangle$

6) Suppose that our qustring has the form $|(00)^2|\hat{3}\hat{x}_1\hat{3}\hat{x}_2\hat{3}\hat{x}_3\rangle|\hat{3}\hat{b}\rangle$ by ignoring $|0^41\rangle$. We want to change the leftmost $\hat{3}$ to $\hat{2}$, resulting in $|(00)^2|\hat{2}\hat{x}_1\hat{3}\hat{x}_2\hat{3}\hat{x}_3\rangle|\hat{3}\hat{b}\rangle$. For our purpose, we first choose a unique bijection p satisfying that $p(\hat{2}) = \hat{3}$, $p(\hat{3}) = \hat{2}$, and $p(y) = y$ for all other $y \in \{0, 1\}^2$. Using Lemma 3.7, we expand this bijection p to its associated quantum function g_p . The quantum function f_6 is then defined as

$$f_6(|\phi\rangle) = \begin{cases} |\phi\rangle & \text{if } \ell(|\phi\rangle) \leq 2, \\ g_p(|00\rangle \otimes f_6(\langle 00|\phi\rangle) + \sum_{y \in \{0, 1\}^2 - \{00\}} |y\rangle\langle y|\phi\rangle) & \text{otherwise.} \end{cases}$$

7) We then change the series of 00's in $|(00)^2|\hat{2}\hat{x}_1\hat{3}\hat{x}_2\hat{3}\hat{x}_3\rangle|\hat{3}\hat{b}\rangle$ into 10's. To make this change, we prepare $h_6 = SWAP \circ NOT \circ CNOT \circ NOT$. Note that $h_6(|00\rangle) = |10\rangle$ and $h_6(|11\rangle) = |11\rangle$. We then set

$$f_7(|\phi\rangle) = \begin{cases} |\phi\rangle & \text{if } \ell(|\phi\rangle) \leq 2, \\ h_6(|00\rangle \otimes f_7(\langle 00|\phi\rangle) + \sum_{y:|y|=2 \wedge y \neq 00} |y\rangle\langle y|\phi\rangle) & \text{otherwise.} \end{cases}$$

This quantum function f_7 changes $|(00)^2\rangle$ to $|(10)^2\rangle$, which equals $|\hat{3}\hat{b}\rangle$, and thus we obtain $|\hat{3}\hat{b}\rangle|\hat{2}\hat{x}_1\hat{3}\hat{x}_2\hat{3}\hat{x}_3\rangle|\hat{3}\hat{b}\rangle$. To obtain $p(n)$ copies of $|\hat{3}\hat{b}\rangle$ in the left-side region of $|\hat{2}\rangle$ in general, the string $(00)^{2p(n)}$ is needed to consume.

8) In this final step, we include the term $|q_0\rangle$ ($= |0^\ell\rangle$) into our procedure. We combine Steps 1)–7) to transform $|0^\ell\rangle|0^{2p(n)-n+1}1\rangle|0^{8p(n)-n+2}1\rangle|x\rangle$ to $|q_0\rangle|\hat{3}\hat{b}\cdots\hat{3}\hat{b}\rangle|\hat{2}\hat{x}_1\hat{3}\hat{x}_2\hat{3}\hat{x}_3\cdots\hat{3}\hat{x}_n\rangle|\hat{3}\hat{b}\cdots\hat{3}\hat{b}\rangle$ by the quantum function F_1 defined as $F_1 = RevBranch_\ell[\{g_s\}_{s \in \{0, 1\}^\ell}]$, where $g_{0^\ell} = f_7$ and $g_s = I$ for any string s different from 0^ℓ .

5.3 Simulating a Single Step

To simulate an entire computation of M on any given input x , we need to simulate all steps of M one by one until M eventually enters the final inner state q_f . In what follows, we demonstrate how to simulate a single step of M by changing a head position, a tape symbol, and an inner state in a given skew configuration.

Note that M 's step involves only three consecutive cells, one of which is being scanned by the tape head. To describe three consecutive cells together with M 's inner state, in general, we use an *expression* r of the form $ps_1\sigma_1s_2\sigma_2s_3\sigma_3$ using $p \in \{0, 1\}^\ell$, $\sigma_i \in \{0, 1, b\}$, and $s_i \in \{2, 3\}$ for each index $i \in [3]$. Each expression with $s_i = 2$ indicates that M is in state q , scanning the i th cell of the three cells. Note that the length of r is $\ell + 6$. Let T be the set of all possible such r 's. Notice that T is a finite set. The *code* $|\hat{r}\rangle$ of $r = ps_1\sigma_1s_2\sigma_2s_3\sigma_3$ is $|q\rangle|\hat{s}_1, \hat{\sigma}_1\rangle|\hat{s}_2, \hat{\sigma}_2\rangle|\hat{s}_3, \hat{\sigma}_3\rangle$, which is of length $\ell + 12$.

For simplicity, we call r a *target* if $s_1 = 3$, $s_2 = 2$, $s_3 = 3$, and $\delta(q, \sigma_2)$ is defined. For later use, $s_1\sigma_1s_2\sigma_2s_3\sigma_3$ without q is called a *pre-target* if $s_1\sigma_1s_2\sigma_2s_3\sigma_3$ is a target.

1) Let $r = qs_1\sigma_1s_2\sigma_2s_3\sigma_3$ be any fixed element in T . We prepare a flag qubit $|0\rangle$ in the end of $|\phi\rangle$ to mark that a simulation of M 's single step is in progress or has been already done. Let us define a quantum function f_8 , which transforms $|\tilde{r}\rangle|0\rangle$ to either $|\tilde{r}\rangle|0\rangle$ or $|\tilde{s}\rangle|1\rangle$, where s is an expression obtained from r by applying δ once if r is a target. Let A denote the set of all targets in T and set $A^\diamond = \{\tilde{r} \mid r \in A\}$. Similarly, we define T^\diamond from T .

For this purpose, we first define a supporting quantum function $h_r(|b\rangle|\psi\rangle) = NOT(|b\rangle) \otimes |\tilde{r}\rangle\langle\tilde{r}|\psi\rangle + \sum_{s \in \{0, 1\}^{\ell+12-A^\diamond}} (|b\rangle \otimes |s\rangle\langle s|\psi\rangle)$ for any target $r \in T$ and any $b \in \{0, 1\}$. Next, we define $\{g_r\}_{r \in T}$ as follows. If r is a non-target in T , then we set $g_r = I$. In what follows, we assume that r is a target. Let $g_r(|0\rangle|s\rangle|\phi\rangle) = |0\rangle|s\rangle|\phi\rangle$ for any string $s \in \{0, 1\}^{\ell+12}$. Since M is plain, M has only two kinds of transitions, shown in (i) and (ii) below.

(i) Consider the case where $\delta(q, \sigma_2) = e^{i\theta}|q', \tau, d\rangle$. When $d = L$, we set

$$g_r(|1\rangle|q\rangle|\hat{3}, \hat{\sigma}_1\rangle|\hat{2}, \hat{\sigma}_2\rangle|\hat{3}, \hat{\sigma}_3\rangle|\phi\rangle) = e^{i\theta}|1\rangle|q'\rangle|\hat{2}, \hat{\sigma}_1\rangle|\hat{3}, \hat{\tau}\rangle|\hat{3}, \hat{\sigma}_3\rangle|\phi\rangle.$$

When $d = R$, in contrast, we define

$$g_r(|1\rangle|q\rangle|\hat{3}, \hat{\sigma}_1\rangle|\hat{2}, \hat{\sigma}_2\rangle|\hat{3}, \hat{\sigma}_3\rangle|\phi\rangle) = e^{i\theta}|1\rangle|q'\rangle|\hat{3}, \hat{\sigma}_1\rangle|\hat{3}, \hat{\tau}\rangle|\hat{2}, \hat{\sigma}_3\rangle|\phi\rangle.$$

(ii) Consider the case where $\delta(q, \sigma_2) = \cos\theta|q_1, \tau_1, d_1\rangle + \sin\theta|q_2, \tau_2, d_2\rangle$. If $(d_1, d_2) = (R, L)$, then we define g_r as

$$g_r(|1\rangle|q\rangle|\hat{3}, \hat{\sigma}_1\rangle|\hat{2}, \hat{\sigma}_2\rangle|\hat{3}, \hat{\sigma}_3\rangle|\phi\rangle) = \cos\theta|1\rangle|q_1\rangle|\hat{3}, \hat{\sigma}_1\rangle|\hat{3}, \hat{\tau}_1\rangle|\hat{2}, \hat{\sigma}_3\rangle|\phi\rangle + \sin\theta|1\rangle|q_2\rangle|\hat{2}, \hat{\sigma}_1\rangle|\hat{3}, \hat{\tau}_2\rangle|\hat{3}, \hat{\sigma}_3\rangle|\phi\rangle.$$

The other values of (d_1, d_2) are similarly handled.

Notice that $\{g_r(|1\rangle|\tilde{r}\rangle)\}_{r \in A}$ forms an orthonormal set because M is well-formed, and thus δ satisfies all the conditions stated in Section 2.2. Once the flag qubit becomes $|1\rangle$, we do not need to apply $g_r \circ h_r$. Thus, we further set $g'_r = Branch[g_r \circ h_r, I]$. By combining all g'_r 's, we define g as $g = Compo[\{g'_r\}_{r \in T^\diamond}]$, which implies

$$g(|0\rangle|\phi\rangle) = \sum_{r \in T} (g_r(|1\rangle|\tilde{r}\rangle\langle\tilde{r}|\phi\rangle)) + \sum_{s \in \{0, 1\}^{\ell+12-T^\diamond}} (|0\rangle|s\rangle\langle s|\phi\rangle)$$

for any quantum state $|\phi\rangle \in \mathcal{H}_\infty$. We can claim that g is norm-preserving and it can be constructed from the initial quantum functions in Definition 3.1 by applying the construction rules. From this claim, g falls in $\widehat{\square_1^{QP}}$. Finally, we define $f_8 = REMOVE_1 \circ (g^{\leq \ell+13} \otimes I)$, where $g^{\leq \ell+13} \otimes I$ is defined as in Lemma 3.8. Intuitively, f_8 changes the content of three consecutive tape cells whose middle cell is being scanned by a tape head.

2) We want to apply f_8 to all three consecutive tape cells. Firstly, we find a code of a pre-target $\hat{s}_1\hat{\sigma}_1\hat{s}_2\hat{\sigma}_2\hat{s}_3\hat{\sigma}_3$ (using the quantum recursion), move an inner state q as well as a marker $|0\rangle$ forward (by $REP_{\ell+1}$) to generate a block of the form $|0\rangle|q\rangle|\hat{s}_1\hat{\sigma}_1\hat{s}_2\hat{\sigma}_2\hat{s}_3\hat{\sigma}_3\rangle$ and apply f_8 to this block. This changes $|0\rangle$ to $|1\rangle$ to record the execution of the current procedure. We then move the obtained inner state and the marker back to the end (by $REMOVE_{\ell+1}$). This entire procedure can be executed by an appropriate quantum function, say, F_2 .

To be more formal, we first set another quantum function p to be $REMOVE_{\ell+1} \circ LENGTH_{\ell+13}[f_8] \circ REP_{\ell+1}$. The quantum function F'_2 is then defined as

$$F'_2(|\phi\rangle|0\rangle|q\rangle) = \begin{cases} |\phi\rangle|0\rangle|q\rangle & \text{if } \ell(|\phi\rangle) < \ell + 13, \\ p(\sum_{s:|s|=4} (|s\rangle \otimes F'_2(\langle s|\phi\rangle|0\rangle|q\rangle))) & \text{otherwise.} \end{cases}$$

To complete the transformation, we further define $F_2 = REP_{\ell+1} \circ F'_2 \circ REMOVE_{\ell+1}$. After the application of F_2 , the first qubit of $|0\rangle|q\rangle|\phi\rangle$ turns to $|1\rangle$, marking an execution of M 's single step. If we want to repeat an application of F_2 , we need to reset $|1\rangle$ back to $|0\rangle$.

5.4 Completing the Entire Simulation

We have shown in Section 5.3 how to simulate a single step of M on x by F_2 . Here, we want to simulate all steps of M by applying F_2 inductively to any input of the form $|0^{p(|x|)}1\rangle \otimes |\psi\rangle$, where $|\psi\rangle$ represents a superposition of coded skew configurations of M on x . This process is implemented by a new quantum function F_3 , which repeatedly applies $NOT \circ F_2$ to $|0\rangle|\psi\rangle$, $p(|x|)$ times, where NOT resets $|1\rangle$ to $|0\rangle$.

We first take a quantum function $\hat{g} = Skip[NOT \circ F_2]$ by Using Lemma 3.9. The desired quantum function F_3 must satisfy

$$F_3(|\phi\rangle) = \begin{cases} |\phi\rangle & \text{if } \ell(|\phi\rangle) \leq 1, \\ |0\rangle \otimes \hat{g}(F_3(\langle 0|\phi\rangle)) + |1\rangle \otimes I(\langle 1|\phi\rangle) & \text{otherwise.} \end{cases}$$

Note that the number of the applications of \hat{g} is exactly $p(|x|)$. This function F_3 can be realized with a use of the single-qubit quantum recursion of the form $F_3 = QRec_1[I, Branch[\hat{g}, I], I|F_3, I]$.

5.5 Preparing an Output

Assume that a superposition of coded skew final configurations of M on the given input x of length n has the form $\sum_{r \in FSC_M(x)} |\widetilde{M[r]}\rangle |\xi_{x,r}\rangle$ for a certain series $\{|\xi_{x,r}\rangle\}_{r \in FSC_{M,n}}$ of quantum states, where $M[r]$ indicates M 's output string appearing in a skew final configuration r including only an essential tape region of M on x .

To show the first part of Lemma 4.3, in the end of the simulation, we need to generate $\widetilde{M[r]}$ in the leftmost portion of the qustring obtained by the simulation. To achieve this goal, we first move the content of the left-side region of the start cell to the right end of the essential tape region.

As an illustrative example, suppose that we have already obtained the quantum state $|\hat{3}\hat{b}\hat{3}\hat{b}\hat{3}\hat{b}\rangle |\hat{2}\hat{0}\hat{3}\hat{0}\hat{3}\hat{1}\rangle |\hat{3}\hat{b}\rangle$ after executing steps in Section 5.4. In this case, the outcome of M in this skew final configuration r is 001, and thus $\widetilde{M[r]} = \hat{0}\hat{0}\hat{1}\hat{2}$ holds. In what follows, we want to transform $|\hat{3}\hat{b}\hat{3}\hat{b}\hat{3}\hat{b}\rangle |\hat{2}\hat{0}\hat{3}\hat{0}\hat{3}\hat{1}\rangle |\hat{3}\hat{b}\rangle$ into $|\hat{0}\hat{0}\hat{1}\hat{2}\rangle |\hat{3}\hat{b}\hat{3}\hat{b}\hat{3}\hat{b}\rangle |\hat{1}\hat{3}\hat{3}\hat{3}\rangle$, which equals $|\widetilde{M[r]}\rangle |\hat{3}\hat{b}\hat{3}\hat{b}\hat{3}\hat{b}\rangle |\hat{1}\hat{3}\hat{3}\hat{3}\rangle$, by an appropriate \square_1^{QP} -function, say, F_4 .

(i) Starting with $|\hat{3}\hat{b}\hat{3}\hat{b}\hat{3}\hat{b}\rangle |\hat{2}\hat{0}\hat{3}\hat{0}\hat{3}\hat{1}\rangle |\hat{3}\hat{b}\rangle$, to mark the end portion of the tape cell, we change the last marker $\hat{3}$ to $\hat{1}$ by applying NOT to $\hat{3}$ and then obtain $|\hat{3}\hat{b}\hat{3}\hat{b}\hat{3}\hat{b}\rangle |\hat{2}\hat{0}\hat{3}\hat{0}\hat{3}\hat{1}\rangle |\hat{1}\hat{b}\rangle$. This process is referred to as f_9 .

(ii) By moving repeatedly the leftmost $|\hat{3}\hat{b}\rangle$ in $|\hat{3}\hat{b}\hat{3}\hat{b}\hat{3}\hat{b}\rangle |\hat{2}\hat{0}\hat{3}\hat{0}\hat{3}\hat{1}\rangle |\hat{1}\hat{b}\rangle$ to the end of of the qustring, we eventually produce $|\hat{2}\hat{0}\hat{3}\hat{0}\hat{3}\hat{1}\rangle |\hat{1}\hat{b}\hat{3}\hat{b}\hat{3}\hat{b}\hat{3}\hat{b}\rangle$. To realize this entire transform, we need to define

$$f_{10}(|\phi\rangle) = \begin{cases} |\phi\rangle & \text{if } \ell(|\phi\rangle) < 4, \\ \sum_{a \in \{0,1\}} |\hat{2}\hat{a}\rangle \langle \hat{2}\hat{a}| \phi \rangle + \sum_{z \in B_4} REMOVE_4(|z\rangle \otimes f_{10}(\langle z|\phi\rangle)) & \text{otherwise,} \end{cases}$$

where $B_4 = \{0,1\}^4 - \{\hat{2}\hat{0}, \hat{2}\hat{1}\}$. More formally, we define f_{11} as $f_{11} = 2QRec_4[I, h'', I|\{f_z\}_{z \in \{0,1\}^4}]$, where $h'' = Branch[\{h''_z\}_{z \in \{0,1\}^4}]$ with $h''_{\hat{2}\hat{0}} = h''_{\hat{2}\hat{1}} = I$ and $h''_z = REMOVE_4$ as well as $f_{\hat{2}\hat{0}} = f_{\hat{2}\hat{1}} = I$ and $f_z = f_{10}$ for any $z \in B_4$.

(iii) The current qustring has the form $|\hat{2}\hat{0}\hat{3}\hat{0}\hat{3}\hat{1}\rangle |\hat{1}\hat{b}\rangle |\hat{3}\hat{b}\hat{3}\hat{b}\hat{3}\hat{b}\rangle$. We sequentially remove each of the markers $\{\hat{1}, \hat{2}, \hat{3}\}$ in $|\hat{2}\hat{0}\hat{3}\hat{0}\hat{3}\hat{1}\rangle |\hat{1}\hat{b}\rangle$ to the end of the entire qustring and produce $|\hat{0}\hat{0}\hat{1}\hat{b}\rangle |\hat{3}\hat{b}\hat{3}\hat{b}\hat{3}\hat{b}\rangle |\hat{1}\hat{3}\hat{3}\hat{2}\rangle$. To implement this process, we apply h_8 defined by $h_8 = 2QREC_3[I, REMOVE_2, I|\{h_y\}_{y \in \{0,1\}^4}]$, where $h_{\hat{1}\hat{b}} = I$ and $h_y = h_8$ for all $y \in \{0,1\}^4 - \{\hat{1}\hat{b}\}$; in other words,

$$h_8(|\phi\rangle) = \begin{cases} |\phi\rangle & \text{if } \ell(|\phi\rangle) < 4, \\ REMOVE_2(|\hat{1}\hat{b}\rangle \langle \hat{1}\hat{b}| \phi \rangle + \sum_{y \in \{0,1\}^4 - \{\hat{1}\hat{b}\}} |y\rangle \otimes h_8(\langle y|\phi\rangle)) & \text{otherwise.} \end{cases}$$

(iv) At last, we change the rightmost \hat{b} in $|\hat{0}\hat{0}\hat{1}\hat{b}\rangle$ to $\hat{2}$ and then obtain $|\hat{0}\hat{0}\hat{1}\hat{2}\rangle (= |\widetilde{M[r]}\rangle)$. To produce such a qustring, we apply the quantum function h_6 defined by

$$h_9(|\phi\rangle) = \begin{cases} |\phi\rangle & \text{if } \ell(|\phi\rangle) < 2, \\ \sum_{y \in \{\hat{0}, \hat{1}\}} |y\rangle \otimes h_9(\langle y|\phi\rangle) + |\hat{b}\rangle \langle \hat{2}| \phi \rangle + |\hat{2}\rangle \langle \hat{b}| \phi \rangle & \text{otherwise.} \end{cases}$$

Formally, we set $h_9 = 2QRec_1[I, h'_9, I|\{h_y\}_{y \in \{0,1\}^2}]$, where $h'_9 = Branch[\{h''_y\}_{y \in \{0,1\}^2}]$ with $h''_{\hat{2}} = h''_{\hat{b}} = SWAP \circ NOT \circ SWAP$ and $h''_y = I$ together with $h_{\hat{2}} = h_{\hat{b}} = I$ and $h_y = h_9$ for any $y \in \{\hat{0}, \hat{1}\}$.

(vi) To perform Steps (i)–(v) at once, we combine all quantum functions used in Steps (i)–(v) and define a single quantum function $F_4 = h_9 \circ h_8 \circ f_{10} \circ f_9$. Overall, the resulted qustring is $|\hat{0}\hat{0}\hat{1}\hat{2}\rangle|\hat{3}\hat{b}\hat{3}\hat{b}\hat{3}\hat{b}\rangle|\hat{1}\hat{3}\hat{3}\hat{2}\rangle$.

The quantum state $|\phi_{F_4}^p(x)\rangle$ can be expressed as $\sum_{r \in FSC_{M,n}} |\widetilde{M[r]}\rangle|\widehat{\xi}_{x,r}\rangle$ for certain quantum states $\{|\widehat{\xi}_{x,r}\rangle\}_r$. Since we deal with an essential tape region of M , it instantly follows that, for every $x, x' \in \{0,1\}^n$ and $r, r' \in FSC_{M,n}$, $\|\langle\xi_{x,r}|\xi_{x',r'}\rangle\| = \|\langle\widehat{\xi}_{x,r}|\widehat{\xi}_{x',r'}\rangle\|$ and $\langle\widehat{\xi}_{x,r}|\widehat{\xi}_{x',r'}\rangle = 0$ if $r \neq r'$. Therefore, F_4 satisfies the condition of the first part of the lemma.

For the second part of the lemma, we further need to retrieve $|M[r]\rangle$ from the coded qustring $|\widetilde{M[r]}\rangle$. From the previous illustrative example $|\hat{0}\hat{0}\hat{1}\hat{2}\rangle|\hat{3}\hat{b}\hat{3}\hat{b}\hat{3}\hat{b}\rangle|\hat{1}\hat{3}\hat{3}\hat{2}\rangle$, we need to produce $|001\rangle|1\rangle|\hat{3}\hat{b}\hat{3}\hat{b}\hat{3}\hat{b}\rangle|\hat{1}\hat{3}\hat{3}\hat{2}\rangle|1000\rangle$. For this purpose, we define the quantum function g_7 by setting $g_7 = 2QRec_1[I, REMOVE_1, I|\{g'_z\}_{z \in \{0,1\}^2}]$; namely,

$$g_7(|\phi\rangle) = \begin{cases} |\phi\rangle & \text{if } \ell(|\phi\rangle) < 2, \\ REMOVE_1(\sum_{y \in \{0,1\}} (|0y\rangle \otimes g_7(\langle 0y|\phi\rangle) + |1y\rangle \langle 1y|\phi\rangle)) & \text{otherwise.} \end{cases}$$

Using g_7 , we finally set $F_5 = g_7 \circ F_4$ to obtain the second part of the lemma.

This completes the proof of Lemma 4.3.

5.6 Simple Applications of the Simulation Procedures

The proof of Lemma 4.3 provides useful procedures not only for the construction of the desired quantum function g but also for other special-purpose quantum functions. Hereafter, as simple applications of the simulation procedures given in Steps 1)–6) of Section 5.2, we will explain how to encode/decode classical strings and how to duplicate classical information by \square_1^{QP} -functions.

Steps 1)–8) in Section 5.2 describe a transformation between classical strings and their encodings. A slight modification of Step 2) introduces an encoder *Encode*, which properly encodes binary strings s to \tilde{s} .

Lemma 5.1 *There exists a quantum function *Encode* in \square_1^{QP} that satisfies $\text{Encode}(|0^{k+1}1\rangle \otimes |\phi\rangle) = \sum_{s \in \{0,1\}^k} (|\tilde{s}\rangle \otimes |s\rangle\langle s|\phi\rangle)$ for any number $k \in \mathbb{N}$ and any quantum state $|\phi\rangle \in \mathcal{H}_\infty$. Moreover, the quantum function *Decode* = Encode^{-1} is also in \square_1^{QP} .*

Given extra bits 0^k1 , the first k qubits of $|\phi\rangle$ is properly encoded by *Encode* by consuming 0^k1 . For example, *Encode* changes $|0^31\rangle|a_1a_2\rangle$ to $|\hat{a}_1\hat{a}_2\hat{2}\rangle$ and *Decode* returns $|\hat{a}_1\hat{a}_2\hat{2}\rangle$ back to $|0^31\rangle|a_1a_2\rangle$. Notice that Proposition 3.5 ensures $\text{Decode} \in \square_1^{\text{QP}}$ from $\text{Encode} \in \square_1^{\text{QP}}$.

Quantum mechanics in general prohibits us from duplicating unknown quantum states; however, it is possible to copy each classical string quantumly. We thus have the following quantum function *COPY*₂, which copies the content of the first k qubits of any input.

Lemma 5.2 *There exists a quantum function *COPY*₂ in \square_1^{QP} that satisfies the following condition: for any $k \in \mathbb{N}^+$ and any $|\phi\rangle \in \mathcal{H}_\infty$,*

$$\text{COPY}_2(|\tilde{0}^k\rangle \otimes |\phi\rangle) = \sum_{s \in \{0,1\}^k} (|\tilde{s}\rangle \otimes |\tilde{s}\rangle\langle \tilde{s}|\phi\rangle).$$

The encoding $\tilde{0}^k$ of 0^k is needed to distinguish 0^k from any part of $|\phi\rangle$ because k is not a fixed constant. When $|\phi\rangle$ has the form $|x\rangle|\psi\rangle$, the quantum function *COPY*₂ works as $\text{COPY}_2(|\tilde{0}^k\rangle \otimes |\tilde{x}\rangle|\psi\rangle) = |\tilde{x}\rangle \otimes |\tilde{x}\rangle|\psi\rangle$.

Proof of Lemma 5.2. We wish to construct the desired quantum function *COPY*₂ as follows. To make our construction process readable, we use an illustrative example of $|\tilde{0}^k\rangle|\phi\rangle = |\tilde{0}^2\rangle|\widetilde{a_1a_2}\rangle (= |\hat{0}\hat{0}\hat{2}\rangle|\hat{a}_1\hat{a}_2\hat{2}\rangle)$ to show how each constructed quantum function works.

1) Steps 1)–6) of Section 5.2 transform $|\hat{0}\hat{0}\hat{1}\rangle|\hat{a}_1\hat{a}_2\rangle$ to $|\hat{2}\hat{a}_1\hat{3}\hat{a}_2\rangle|\hat{3}\rangle$. By a slight modification of these steps, it is possible to transform $|\hat{0}\hat{0}\hat{2}\rangle|\hat{a}_1\hat{a}_2\hat{2}\rangle$ to $|\hat{3}\hat{a}_1\hat{3}\hat{a}_2\hat{2}\rangle|\hat{2}\rangle$. We denote by f_1 a quantum function that realizes this transformation.

2) By copying \hat{a}_i in $\hat{3}\hat{a}_i$ onto $\hat{3}$ for each $i \in \{1, 2\}$, we change $\hat{3}\hat{a}_i$ to $\hat{a}_i\hat{a}_i$ and then obtain $|\hat{a}_1\hat{a}_1\hat{a}_2\hat{a}_2\hat{2}\rangle|\hat{2}\rangle$. This step can be formally made in the following way. Firstly, we define a quantum function h_2 that satisfies $h_2(|b_1b_2\rangle|b_3b_4\rangle \otimes |\phi\rangle) = |b_4b_2b_1b_3\rangle \otimes |\phi\rangle$ for any $\phi \in \mathcal{H}_\infty$. Such a quantum function actually exists by Lemma 3.7. Secondly, we set *DUP* to be $h_2^{-1} \circ (SWAP \circ NOT \circ SWAP \circ CNOT) \circ h_2 \circ NOT$. It then follows that

$DUP(|\hat{3}\rangle|\hat{a}\rangle \otimes |\phi\rangle) = |\hat{a}\rangle|\hat{a}\rangle \otimes |\phi\rangle$ for any bit $a \in \{0, 1\}$ and any quantum state $|\phi\rangle \in \mathcal{H}_\infty$. With this DUP , we further define f_2 by the 4-qubit quantum recursion as

$$f_2(|\phi\rangle) = \begin{cases} |\phi\rangle & \text{if } \ell(|\phi\rangle) < 4, \\ DUP(\sum_{a \in \{0, 1\}} |\hat{3}\hat{a}\rangle \otimes f_2(|\hat{3}\hat{a}\rangle)) + \sum_{y \in B'_4} |y\rangle\langle y|\phi\rangle & \text{otherwise,} \end{cases}$$

where $B'_4 = \{0, 1\}^4 - \{\hat{3}\hat{0}, \hat{3}\hat{1}\}$.

3) Next, we transform $|\hat{a}_1\hat{a}_1\hat{a}_2\hat{a}_2\hat{2}\rangle|\hat{2}\rangle$ to $|\hat{a}_1\hat{a}_2\hat{2}\rangle|\hat{2}\hat{a}_2\hat{a}_1\rangle$. This transformation can be done by f_3 defined as

$$f_3(|\phi\rangle) = \begin{cases} |\phi\rangle & \text{if } \ell(|\phi\rangle) < 4, \\ REMOVE_2(\sum_{y \in \{0, 1\}^4 - \{\hat{2}\hat{2}\}} |y\rangle \otimes f_3(|y\rangle)) + |\hat{2}\hat{2}\rangle\langle \hat{2}\hat{2}|\phi\rangle & \text{otherwise.} \end{cases}$$

4) We then change $|\hat{a}_1\hat{a}_2\hat{2}\rangle|\hat{2}\hat{a}_2\hat{a}_1\rangle$ to $|\hat{2}\hat{a}_2\hat{a}_1\rangle|\hat{2}\hat{a}_2\hat{a}_1\rangle$ by removing each two qubits in the first part to the end. This process is precisely realized by f_4 defined as

$$f_4(|\phi\rangle) = \begin{cases} |\phi\rangle & \text{if } \ell(|\phi\rangle) < 4, \\ REMOVE_2(\sum_{y \in \{0, 1\}^2 - \{\hat{2}\}} (|y\rangle \otimes f_5(|y\rangle)) + |\hat{2}\rangle\langle \hat{2}|\phi\rangle) & \text{otherwise.} \end{cases}$$

5) Finally, we reverse the whole qustring to obtain $|\hat{a}_1\hat{a}_2\hat{2}\rangle|\hat{a}_1\hat{a}_2\hat{2}\rangle$ by applying $REVERSE$.

This completes the proof of the lemma. \square

6 Future Challenges

In Definition 3.1, we have defined \square_1^{QP} -functions on \mathcal{H}_∞ and we have given in Theorem 4.1 a new characterization of FBQP-functions in terms of these \square_1^{QP} -functions. To point out the directions of future research, we wish to raise a challenging open question in Section 6.1 and to present in Sections 6.2–6.3 three possible implications of our schematic definition to the subjects of descriptional complexity, first-order theories, and higher-type functionals. In Section 6.4, we will remark a practical application to the designing of quantum programming languages.

6.1 Seeking a More Reasonable Schematic Definition

Our schematic definition (Definition 3.1) is composed of the initial quantum functions, which are derived from natural, simple quantum gates, and the construction rules, including the multi-qubit quantum recursion, which significantly enriches the scope of constructed quantum functions. The choice of initial functions and construction rules that we have used in this paper directly affects the richness of \square_1^{QP} -functions. Although our \square_1^{QP} is sufficient to characterize FBQP, if we seek for enriching the \square_1^{QP} -functions, one way is to supplement additional initial quantum functions. As a concrete example, let us consider the quantum Fourier transform (QFT), which plays an important role in, e.g., Shor’s factoring quantum algorithm [30]. We have demonstrated in Lemma 3.10 how to implement a restricted form of QFT working on a fixed number of qubits, and thus it belongs to \square_1^{QP} . Nonetheless, a more general form of QFT, acting on an “arbitrary” number of qubits, may not be realized *precisely* by \square_1^{QP} -functions although it can be approximated to any desired accuracy by the \square_1^{QP} -functions. To remedy the exclusion of QFT from our function class \square_1^{QP} , for instance, we can expand the current \square_1^{QP} by including as an initial quantum function the quantum function defined as

$$CROT(|\phi\rangle|0^j\rangle) = |0\rangle\langle 0|\phi\rangle|0^j\rangle + \omega_j|1\rangle\langle 1|\phi\rangle|0^j\rangle \text{ (controlled rotation),}$$

where $\omega_j = e^{2\pi i/2^j}$, with an extra term 0^j . It is not difficult to construct QFT from quantum functions in this expanded $\widehat{\square_1^{QP}}$ obtained by adding $CROT$. As this example shows, it remains important to seek for a simpler, more reasonable schematic definition of quantum functions, which are capable of precisely characterizing both BQP and FBQP and also simplifying the proof of Theorem 4.1.

From the minimalist’s viewpoint, on the contrary, we may be able to eliminate certain schemata or replace them by simpler ones but still ensure the characterization result of BQP and FBQP in terms of \square_1^{QP} -functions. As a concrete example, we may ask whether our multi-qubit quantum recursion can be replaced by 1-qubit quantum recursion at the cost of adding extra initial quantum functions.

6.2 Introduction of Descriptive Complexity and First-Order Theories

As noted in Section 3.1, our schematic definition of \square_1^{QP} -functions provides us with a natural means of assigning the *descriptive complexity*—a new complexity measure—to each of those quantum functions in \square_1^{QP} . This complexity measure has been used to prove, for instance, Lemma 3.4. As a consequence of our main theorem, this complexity measure concept also transfers to languages in BQP and functions in FBQP, and thus it naturally helps us introduce the notion of the descriptive complexities of such languages and functions.

It is further possible for us to extend this complexity measure to “arbitrary” languages and functions on $\{0, 1\}^*$, which are not necessarily limited to FBQP and BQP, and to discuss their “relative” complexity to \square_1^{QP} . More formally, given a function f on $\{0, 1\}^*$, the \square_1^{QP} -*descriptive complexity* of f at length n is the minimal number of times we use initial quantum functions and construction rules to build a \square_1^{QP} -function g for which $\ell(|\phi_g^p(x)\rangle) = |f(x)|$ and $|\langle f(x)|\phi_g^p(x)\rangle|^2 \geq 2/3$ hold for a certain polynomial p with $|f(x)| \leq p(|x|)$ for all strings x of length exactly n . We write $dc(f)[n]$ to denote the \square_1^{QP} -descriptive complexity of f at length n . Obviously, every \square_1^{QP} -function has *constant* \square_1^{QP} -descriptive complexity at every length. In a similar spirit but based on quantum finite automata, Villagra and Yamakami [33] discussed the quantum state complexity restricted to inputs of length exactly n (as well as length at most n). It has turned out that such complexity measure is quite useful. Refer to [33] for the detailed definitions. Our new complexity measure $dc(f)[n]$ is also expected to be a useful tool in classifying languages and functions in descriptive power in a way that is quite different from what QTMs and quantum circuits do.

In a much wider perspective, our schematic definition of polynomial-time quantum computability may lead to the future development of an appropriate form of first-order theories over quantum states in Hilbert spaces or *first-order quantum theories*, for short. In the literature, first-order theories and their natural subtheories have become a fruitful research subject in mathematical logic and recursion theory and they have also found numerous applications in other fields as well. A weak form of their subtheories has been studied in quantum complexity theory. For instance, using bounded quantifiers over quantum states in Hilbert spaces, quantum analogues of NP and the Meyer-Stockmeyer polynomial(-time) hierarchy have been discussed in [37]. Unfortunately, we are still far away from obtaining well-accepted first-order theories and useful subtheories for quantum computing.

6.3 Extension to Type-2 Quantum Functionals

Conventionally, functions mapping Σ^* to Σ^* are categorized as *type-1 functionals*, whereas *type-2 functionals* are functions taking inputs from Σ^* together with type-1 functionals. In computational complexity theory, such *type-2 functionals* have been extensively discussed in, e.g., [8, 9, 24, 31, 35].

In analogy to the classical case, we call \square_1^{QP} -functions on \mathcal{H}_∞ *type-1 quantum functionals*. To introduce *type-2 quantum functionals*, we start with an arbitrary quantum function O mapping \mathcal{H}_∞ to \mathcal{H}_∞ , which is treated as a *function oracle* (an *oracle function* or simply an *oracle*) in the following formulation. From such an oracle O , we define a new linear operator \tilde{O} as $\tilde{O}(|\tilde{x}\rangle) = \sum_{s:|s|=|x|} \alpha_s |\tilde{s}\rangle$ if $O(|x\rangle)$ is of the form $\sum_{s:|s|=|x|} \alpha_s |s\rangle$ for any string $x \in \{0, 1\}^*$, where \tilde{s} is a code of s , defined in Section 4.1. Note that $\tilde{O}(|\tilde{x}\rangle) = \widetilde{O}(|x\rangle)$ and $\ell(\tilde{O}(|\tilde{x}\rangle)) = 2\ell(O(|x\rangle)) + 2$.

Firstly, we expand our initial functions and construction rules given in Definition 3.1 by replacing any quantum function, say, $f(|\phi\rangle)$ in each scheme of the definition with $f(|\phi\rangle, O)$. Secondly, we introduce another initial quantum function, called the *query function* *QUERY*. For any two qustrings $|\phi\rangle$ and $|\psi\rangle$ of length n , let $|\phi\rangle \oplus |\psi\rangle = \sum_{s:|s|=n} \sum_{t:|t|=n} \langle s|\phi\rangle \langle t|\psi\rangle |s \oplus t\rangle$, where $s \oplus t$ means the *bitwise XOR* of s and t . As a special case, it follows that $\tilde{O}(|\tilde{x}\rangle) \oplus \tilde{O}(|\tilde{x}\rangle) = |0^{2|x|+2}\rangle$ for any $x \in \{0, 1\}^*$. The query function is then defined as

$$\begin{aligned} & \text{QUERY}(|\phi\rangle, O) \\ &= \sum_{n \in [t]} \sum_{x \in \Sigma^n} \sum_{s \in \Sigma^n} \left(|\tilde{x}\rangle \otimes (\tilde{O}(|\tilde{x}\rangle) \oplus |\tilde{s}\rangle) \otimes \langle \tilde{x}\tilde{s}| \phi \rangle + \sum_{y \in \Sigma^{4n+4} \wedge y \neq \tilde{x}\tilde{s}} |y\rangle \langle y| \phi \rangle \right) \end{aligned}$$

for all $|\phi\rangle \in \mathcal{H}_\infty$, where $t = \lfloor (\ell(|\phi\rangle) - 4)/4 \rfloor$. In particular, $\text{QUERY}(|\tilde{x}\rangle|\tilde{s}\rangle|\phi\rangle)$ equals $|\tilde{x}\rangle \otimes (\tilde{O}(|\tilde{x}\rangle \oplus |\tilde{s}\rangle) \otimes |\phi\rangle)$. It also follows that $\text{QUERY} \circ \text{QUERY}(|\phi\rangle, O) = I(|\phi\rangle)$ since $\tilde{O}(|\tilde{x}\rangle) \oplus \tilde{O}(|\tilde{x}\rangle) = |0^{2|x|+2}\rangle$.

Notice that if O is in \square_1^{QP} then the function $Q_O(|\phi\rangle) =_{\text{def}} \text{QUERY}(|\phi\rangle, O)$ also belongs to \square_1^{QP} . It is also possible to show similar results discussed in the previous sections. These basic results can open a door to a rich field of higher-type quantum computability and we expect fruitful results to be discovered in this new field.

6.4 Application to Quantum Programming Languages

A practical application of our schematic definition can be found in the area of *quantum programming languages*. Since the early days of quantum computing research, a significant effort has been made by physicists, computer scientists, and computer engineers to draw a pragmatic road map to a real-life quantum computer.

Toward the realization of such quantum computers, most research has focused on their hardware construction. For the building of “multi-purpose” quantum computers, however, it is more desirable to make them “programmable” in such a way that a run of an appropriate “quantum program” freely alters computing processes for different target problems without remodeling their hardware each time. A quantum program here refers to a finite series of instructions on how to operate the quantum computer step by step. To write such a quantum program, nevertheless, we need to develop well-structured programming languages for the quantum computer (dubbed as *quantum programming languages*). Various quantum programming languages have been discussed over two decades in due course of developing real-life quantum computers. Refer to surveys, e.g., [13] for a necessary background.

Our schematic definition provides a description of how to define a given \square_1^{QP} -function. This description can be viewed as a set of instructions, each of which instructs how to apply each scheme to construct the desired quantum function and it thus resembles a program that dictates how to construct the quantum function. Therefore, our schematic description of a construction process of quantum functions may help us design appropriate quantum programming languages in the future.

References

- [1] L. M. Adleman, J. DeMarrais, and M. A. Huang, Quantum computability, *SIAM J. Comput.* **26** (1997) 1524–1540.
- [2] A. Barenco, C. H. Bennett, R. Cleve, D. DiVicenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter, Elementary gates for quantum computation, *Phys. Rev. A*, **52** (1995) 3457–3467.
- [3] P. Benioff, The computer as a physical system: a microscopic quantum mechanical Hamiltonian model of computers represented by Turing machines. *J. Statist. Phys.* **22** (1980) 563–591.
- [4] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, Strengths and weaknesses of quantum computing, *SIAM J. Comput.* **26** (1997) 1510–1523.
- [5] E. Bernstein and U. Vazirani, Quantum complexity theory, *SIAM J. Comput.* **26** (1997) 1411–1473.
- [6] P. O. Boykin, T. Mor, M. Pulver, V. Roychowdhury, and F. Vatan, On universal and fault-tolerant quantum computing. arXiv:quant-ph/9906054, 1999.
- [7] A. Cobham, The intrinsic computational difficulty of functions, in the *Proceedings of the 1964 Congress for Logic, Mathematics, and Philosophy of Science*, pp. 24–30, North-Holland, 1964.
- [8] R. L. Constable, Type two computational complexity, in the *Proceedings of the 5th ACM Symposium on Theory of Computing* (STOC’73), pp. 108–121, 1973.
- [9] S. Cook and B. M. Kapron, Characterizations of the basic feasible functionals of finite type, in the *Proceedings of the 30th IEEE Conference on Foundations of Computer Science* (STOC’89), pp. 154–159, 1989.
- [10] M. Davis, (ed.) *The Undecidable. Basic Papers on Undecidable Propositions, Unsolvable Problems, and Computable Functions*, Raven Press, Hewlett, New York, 1965.
- [11] D. Deutsch, Quantum theory, the Church-Turing principle, and the universal quantum computer, *Proceedings Royal Society London*, Ser. A, **400** (1985) 97–117.
- [12] D. Deutsch, Quantum computational networks, *Proceedings Royal Society London*, Ser. A, **425** (1989) 73–90.
- [13] S. J. Gay, Quantum programming languages: survey and bibliography. *Math. Struct. in Comp. Science* **16** (2006) 581–600.
- [14] L. K. Grover, A fast quantum mechanical algorithm for database search, in the *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing* (STOC’96), pp. 212–219, 1996.
- [15] L. Grover, Quantum mechanics in searching for a needle in a haystack. *Physical Review Letters* 79:325 (1997).
- [16] J. Hopcroft and J. Ullman, *An Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, Reading MA, 1979.
- [17] A. Kitaev. Quantum computations: algorithms and error correction. *Russian Math. Surveys* **52** (1997) 1191–1249.
- [18] A. Y. Kitaev, A. H. Shen, and M. N. Vyalyi. *Classical and Quantum Computation* (Graduate Studies in Mathematics), American Mathematical Society, 2002.
- [19] S. C. Kleene, General recursive functions of natural numbers, *Math. Ann.* **112** (1936) 727–742.
- [20] S. C. Kleene, Recursive predicates and quantifiers, *Trans. A. M. S.* **53** (1943) 41–73.