The Hidden Subgroup Problem

Course Project for PHY 612: Quantum Information Science

Name: Alina Zainab Rizvi, Diya Batool, and Hassan Mehmood
IDs: 23100010, 23100267, 23100127

April 2023

Abstract

Let G be a finite abelian group and X be a finite set. Consider a function $f : G \to X$ that is periodic and distinct on each coset in G of a subgroup K of G. Does there exist a quantum algorithm to find this subgroup K? This is called the hidden subgroup problem, and can be regarded as a generalisation of a large class of computational problems that involve finding the period of a periodic function, such as Shor's factoring algorithm, discrete logarithms, order of a permutation, and so on. In this report, we discuss a solution to the hidden subgroup problem, and apply it to some specific instances.

Contents

1 Introduction

If one carefully analyzes some major quantum algorithms, it is discovered that their basic structure consists in finding a solution to a query pertaining to a function f from a finite set A to another finite set B. In general, the action of this function is typically described by means of a certain unitary operator U acting on a set of n quantum registers $|a_1\rangle, \ldots, |a_{n-1}\rangle, |b\rangle$ such that $U|a_1\rangle \cdots |a_n\rangle \rightarrow |a_1\rangle \cdots |a_{n-1}\rangle|b \oplus f(a_1, \ldots, a_{n-1})\rangle$, i.e. the last register stores the effect of applying the function to the preceding registers. The query one wishes to solve depends on the specific problem at hand. For instance, in the Deutsch algorithm, we wish to find the bias of a function on a bit-valued function on an n-bit string. In other problems, such as order-finding, factoring and discrete logarithms, one uses the quantum Fourier transform to estimate the eigenvalues of U, which, owing to the specific form of f, are directly related to the object one seeks, e.g. the order, factor or the discrete logarithm of a positive integer. What unites all of these seemingly disparate problems is the fact that in each case, the function f acts on the domain A in a manner that makes it possible to abstract from these specific problems to a much more general problem using tools from group theory. For instance, in the Deutsch algorithm, the bias of the function depends on how the function acts on the subsets of $\{0, 1\}$, while in the other three problems mentioned above, the function is periodic, and what we seek is precisely its period, which, again, depends on how the function acts on different "parts" of its domain. These are all specific instances of a function being distinct and constant on the cosets of a subgroup of a group. Therefore, all these different problems – and many others, as we shall see – can be reduced to the problem of finding that subgroup. This is known as the hidden subgroup problem, which, in the case of a finite abelian group, is fully solvable in a polynomial amount of elementary quantum operations.

2 Mathematical Preliminaries

Before we can even precisely define the hidden subgroup problem, it is necessary to understand the relevant group-theoretic concepts that are used in the formulation of the problem. This section is devoted to that task.

2.1 Groups, subgroups, cosets, and all that

Definition 1. *(Group) A group is an ordered pair $(G, *)$, where $*$ is a binary operation on the set G. The binary operation must satisfy the following axioms:*

1. *associativity : $(a * b) * c = a * (b * c)$*

2. *there exists an element e, called identity, in G such that $a * e = e * a = a$ for all $a \in G$*

3. there exists an inverse $a^{-1} \in G$ for each $a \in G$ such that $a^{-1} * a = a * a^{-1} = e$.

For instance, the set of all invertible square matrices forms a group under matrix multiplication. Often the multiplication operation $*$ is understood from the context, and so we simply denote the group by G, and write $a * b$ as ab for any $a, b \in G$.

If G is a finite set, then G is called a *finite group*, and we denote by $|G|$ the number of elements in G. If the binary operation is commutative, i.e. $a * b = b * a$ for every $a, b \in G$, then the G is said to be *abelian.*

For example, the set of integers $\mathbb{Z}$ form a group under addition $(\mathbb{Z}, +)$. The associativity property holds for this group and identity element $e = 0$ and inverse $a^{-1} = -a$. Furthermore, since addition is commutative, $\mathbb{Z}$ is an abelian group. But this group is, of course, not finite. On the other hand, consider $\mathbb{Z}_6 := \{0, 1, 2, 3, 4, 5\}$, the set of integers defined by addition modulo 6. This is a finite abelian group.

Definition 2. *(Subgroup) A subgroup H of a group G is a nonempty subset of the group G that is closed under inverses and products. That is, for $x, y \in H$, the inverse $x^{-1} \in H$ and $x * y \in H$.*

We write $H \leq G$ to denote that H is a subgroup of G. It follows from the definition above that a subgroup is itself a group. For example, the set $\{0, 1, 5\}$ is a subgroup of $\mathbb{Z}_6$.

Definition 3. *(Group homomorphism) Two groups $(G, \star)$ and $(H, \diamond)$ are homomorphic if there exists a map $\phi : G \to H$ such that $\phi(a \star b) = \phi(a) \diamond \phi(b)$ for all $a, b \in G$.*

Intuitively, two groups have the same group-theoretic structure or 'look the same' if they are homomorphic. If a homomorphism also happens to be a bijection, then it is called an *isomorphism.* Isomorphic groups are completely equivalent in the sense that one can work in any one without loss of generality. If two groups G and F are isomorphic, we write $G \cong F$.

For example, the set of real numbers $\mathbb{R}$ forms a group under addition '$+$'. Similarly, the set of nonnegative real numbers $\mathbb{R}_{\geq 0}$ is a group under ordinary multiplication '$\cdot$'. These two groups are homomorphic, the homomorphism being the exponential function e^x from $\mathbb{R}$ to $\mathbb{R}_{\geq 0}$. Indeed, it is easily seen that $e^{x+y} = e^x \cdot e^y$. Furthermore, if we restrict the additive group to nonnegative real numbers only, then the map is also a bijection, as can be confirmed by drawing a graph of e^x. Thus $(\mathbb{R}_{\geq 0}, +) \cong (\mathbb{R}_{\geq 0}, \cdot)$.

Definition 4. *(Cosets) If H is a subgroup of group G, the left coset of H in G determined by $g \in G$ is the set $gH \equiv \{gh \mid h \in H\}$. The right coset is defined similarly.*

Often whether a coset is a 'left' or 'right' coset is implied by context. In the case of abelian groups, the left coset will be equal to the right coset.

Definition 5. *(Characters) Given a group G, a character is a homomorphism from G to the group of complex numbers $\mathbb{C}$ under ordinary multiplication.*

For instance, the function $f : (\mathbb{R}, +) \to (\mathbb{C}, \cdot)$ such that $f(x) = e^{ix}$ is a homomorphism. Thus it is a character of the additive group of real numbers.

Definition 6. *(Dual group) The dual $\hat{G}$ of a group G is the set of all characters of G.*

The dual of a group is itself a group under functional multiplication, i.e. for any $\alpha, \beta \in \hat{G}$, we define $\alpha \cdot \beta$ by $(\alpha \cdot \beta)(g) = \alpha(g)\beta(g)$ for all $g \in G$.

If G is a finite abelian group, then the number of characters of G is equal to the number of elements in G. In other words, $|\hat{G}| = |G|$.

2.2 Fourier transforms over groups

Given the above definitions – particularly of characters and dual groups – we can finally define Fourier transforms.

Definition 7. *(Fourier Transform) Given the characters of group G, the Fourier transform over a group G is given by*

$$|g\rangle \mapsto \frac{1}{|G|} \sum_{\tilde{g} \in \hat{G}} \tilde{g}(g) |\tilde{g}\rangle$$

Now that we have defined Fourier transforms, the following theorems would help us in solving the hidden subgroup problem.

Theorem 1. *For each subgroup $H \subset G$, let $H^\perp \subseteq \hat{G}$ be such that $H^\perp = \{k \in \hat{G} \mid k(h) = 1$ $\forall h \in H\}$. The Fourier transform over G maps an equal superposition on H to an equal superposition over $H^\perp$:*

$$\frac{1}{|H|} \sum_{h \in H} |h\rangle \mapsto \sqrt{\frac{|H|}{|G|}} \sum_{k \in H^\perp} |k\rangle$$

Proof. Let's consider a state $|\psi\rangle$ such that

$$|\psi\rangle = \frac{1}{|H|} \sum_{h \in H} |h\rangle$$

Doing a Fourier transform over the state $|\psi\rangle$ we get:

$$\begin{aligned} \frac{1}{|H|} \sum_{h \in H} |h\rangle \mapsto & \sqrt{\frac{1}{|H||G|}} \sum_{h \in H} \sum_{\tilde{g} \in \hat{G}} \tilde{g}(h) |\tilde{g}\rangle \\ &= \frac{1}{\sqrt{|H||G|}} \sum_{h \in H} \left[\sum_{\tilde{g} \in H^\perp} \tilde{g}(h) |\tilde{g}\rangle + \sum_{\tilde{g} \notin H^\perp} \tilde{g}(h) |\tilde{g}\rangle \right] \end{aligned}$$

$$= \frac{1}{\sqrt{|H||G|}} \sum_{h\in H} (1) \sum_{\tilde{g}\in H^{\perp}} |\tilde{g}\rangle + \frac{1}{\sqrt{|H||G|}} \sum_{h\in H} \sum_{\tilde{g}\in H^{\perp}} \tilde{g}(h) \, |\tilde{g}\rangle$$

$$= \sqrt{\frac{|H|}{|G|}} \sum_{\tilde{g}\in H} |\tilde{g}\rangle + \frac{1}{\sqrt{|H||G|}} \sum_{h\in H} \sum_{\tilde{g}\in H^{\perp}} \tilde{g}(h) \, |\tilde{g}\rangle \, , \tag{1}$$

where the third line follows from the fact that $\tilde{g}(h) = 1$ for all $\tilde{g} \in H^{\perp}$ and $h \in H$. Thus to establish the theorem, all we need to do is to show that the second term above is zero. We now demonstrate this. To begin with, for each $\tilde{g} \in H^{\perp}$, we have that

$$\alpha_{\tilde{g}} := \frac{1}{\sqrt{|H||G|}} \sum_{h\in H} \tilde{g}(h) = \frac{1}{\sqrt{|H||G|}} \sum_{h\in H} (1) = \sqrt{\frac{|H|}{|G|}} = \frac{1}{\sqrt{|H^{\perp}|}},$$

where the last equality comes from $|G| = |H||H^{\perp}|$. Thus,

$$1 = \sum_{\tilde{g}\in \hat{G}} |\alpha_{\tilde{g}}|^2 = \sum_{\tilde{g}\in H^{\perp}} |\alpha_{\tilde{g}}|^2 + \sum_{\tilde{g}\notin H^{\perp}} |\alpha_{\tilde{g}}|^2 = \frac{1}{|H^{\perp}|} \sum_{\tilde{g}\in H^{\perp}} (1) + \sum_{\tilde{g}\notin H^{\perp}} |\alpha_{\tilde{g}}|^2 = 1 + \sum_{\tilde{g}\notin H^{\perp}} |\alpha_{\tilde{g}}|^2,$$

whence

$$|\alpha_{\tilde{g}}|^2 = 0 \quad \forall \tilde{g} \notin H^{\perp}.$$

and so $\tilde{g}(h) = 0$ for every $\tilde{g} \notin H^{\perp}$ and $h \in H$.

□

Theorem 2. *The Fourier transform over G maps an equal superposition on cosets of H to an equal superposition over cosets of $H^{\perp}$.*

$$\frac{1}{|H|} \sum_{h\in H} |hg\rangle \mapsto \sqrt{\frac{|H|}{|G|}} \sum_{\tilde{h}\in H^{\perp}} \tilde{h}(g) \, |\tilde{h}\rangle \tag{2}$$

Proof. From the fact that $\tilde{g}$ is a homomorphism, we have that $\tilde{g}(hg) = \tilde{g}(h)\tilde{g}(g) \ \forall \tilde{g} \in \hat{G}$ and $g, h \in G$. Doing a Fourier transform on the state $|hg\rangle$, we thus get :

$$\frac{1}{|G|} \sum_{\tilde{g}\in \hat{G}} \tilde{g}(hg) \, |\tilde{g}\rangle = \frac{1}{|G|} \sum_{\tilde{g}\in \hat{G}} \tilde{g}(h)\tilde{g}(g) \, |\tilde{g}\rangle$$

Therefore, we can write the complete Fourier transform as

$$\frac{1}{|H|} \sum_{h\in H} |hg\rangle \mapsto \sum_{\tilde{g}\in \hat{G}} \left[\sqrt{\frac{1}{|H||G|}} \sum_{h\in H} \tilde{g}(h)\tilde{g}(g) \right] |\tilde{g}\rangle \, .$$

As shown in the previous theorem, $\tilde{g}(h) = 0$ for all $\tilde{g} \notin H^{\perp}$ and $h \in H$. Therefore, the preceding sum reduces to

$$\sqrt{\frac{1}{|H||G|}} \sum_{\tilde{g} \in H^{\perp}} \left[\sum_{h \in H} \tilde{g}(h)\tilde{g}(g) \right] |\tilde{g}\rangle = \sqrt{\frac{1}{|H||G|}} \sum_{\tilde{g} \in H^{\perp}} \left[\sum_{h \in H} (1)\tilde{g}(g) \right] |\tilde{g}\rangle$$
$$= \sqrt{\frac{|H|}{|G|}} \sum_{\tilde{g} \in H^{\perp}} \tilde{g}(g) \, |\tilde{g}\rangle \, .$$

□

Notice that since the characters map group elements to complex numbers, it follows from

$$\sum_{\tilde{g} \in \hat{G}} |\tilde{g}(g)|^2 = 1$$

that $\tilde{g}(g)$ is just a phase, which thus has no effect on the probability of measuring $|\tilde{h}\rangle$ in (2). This immediately yields the following theorem.

Theorem 3. *Fourier sampling on an equal superposition on a coset of H will yield a uniformly random element $k \in H$.*

3 The Hidden Subgroup Problem

3.1 Formulation

We are now ready to formulate the hidden subgroup problem (HSP):

> Let G be a finite abelian group and X be a finite set. Suppose that there exists a function $f : G \to X$ that is distinct and constant on each coset of a subgroup H of G. Thus $f(g) = f(g')$ if and only if $g' = hg$ for some $h \in H$. Suppose that we possess a unitary operator U that performs the operation $|g\rangle|x\rangle \to |g\rangle|x \oplus f(g)\rangle$, where $g \in G$, $x \in X$ and $\oplus$ is an appropriately chosen binary operation on X. Find the subgroup H.

As we outlined in the Introduction, many textbook quantum algorithms can be regarded as specific instances of the hidden subgroup problem. For example, consider the problem of finding the period r of a periodic function $f : \mathbb{Z} \to X$, where X is any finite set. In other words, we wish to find r such that $f(z + r) = f(z)$ for all $z \in \mathbb{Z}$. Now $\mathbb{Z}$ is a group under addition '+'. Consider $H = \{0, r, 2r, \ldots\} \leq \mathbb{Z}$. It is not difficult to see that f is constant and distinct on each coset $z + H := \{z, z + r, z + 2r, \ldots\}$ of H. Thus, finding the period r is equivalent to finding the subgroup H. Fig 1 lists several other well-known problems that are specific cases of the hidden subgroup problem; in each case, the relevant groups, functions and subgroups are identified.]

Using the results of the previous section, we can construct a general algorithm to solve the hidden subgroup problem. This we do in the next subsection.

3.2 Implementation

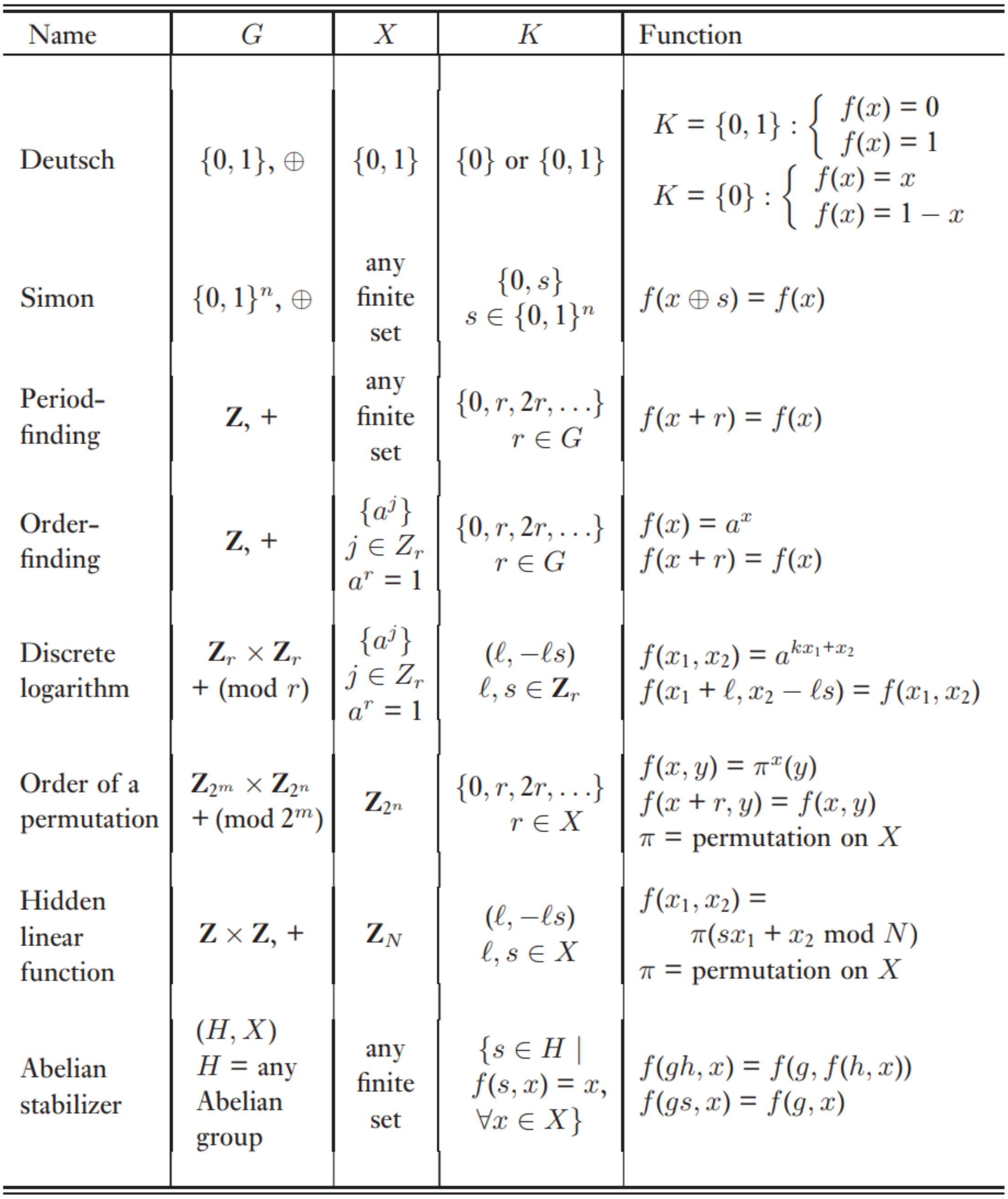
The HSP can be solved using a quantum algorithm that can be divided into steps. This can be summarized as follows [3]:

$$|0\rangle\,|0\rangle \xrightarrow{FT_G} \frac{1}{\sqrt{|G|}}\sum_{a\in G}|a\rangle\,|0\rangle \xrightarrow{f:G\to X} \frac{1}{\sqrt{|G|}}\sum_{a\in G}|a\rangle\,|f(a)\rangle \xrightarrow{measurement} \frac{1}{\sqrt{|H|}}\sum_{h\in H}|hg\rangle\,.$$

First, set up a random coset state, wherein two quantum registers are taken and initialized to $|0\rangle$. Both registers are such that they can store the elements belonging to group G. Once that is done, we take the Fourier transform of the initial quantum register. To this, we apply the function $f : G \to X$. The result of applying these steps on the *first* register is to be stored in the second register, whose measurement is taken. Suppose we find the state of the second register to be $|f(g)\rangle$ for some $g \in G$ after measurement. Since $f(gh) = f(g)$ for all $h \in H$, the state of the first register becomes a uniform superposition over the coset gH.

In the second stage of the algorithm, we take the first register's Fourier transform and measure it. This allows us to obtain constraints on the hidden subgroup H. And what is that constraint? According to the last theorem in the previous section, it is the random element belonging to $H^{\perp}$ that is obtained due to the measurement of the register after its Fourier transformation. These constraints can be solved to find out H.

For future reference, we enlist the essential steps involved the algorithm above below (p. 9).



Name	G	X	K	Function
Deutsch	$\{0,1\}, \oplus$	$\{0,1\}$	$\{0\}$ or $\{0,1\}$	$K = \{0,1\} : \begin{cases} f(x) = 0 \\ f(x) = 1 \end{cases}$ $K = \{0\} : \begin{cases} f(x) = x \\ f(x) = 1 - x \end{cases}$
Simon	$\{0,1\}^n, \oplus$	any finite set	$\{0, s\}$ $s \in \{0,1\}^n$	$f(x \oplus s) = f(x)$
Period-finding	$\mathbf{Z}$, +	any finite set	$\{0, r, 2r, \ldots\}$ $r \in G$	$f(x + r) = f(x)$
Order-finding	$\mathbf{Z}$, +	$\{a^j\}$ $j \in Z_r$ $a^r = 1$	$\{0, r, 2r, \ldots\}$ $r \in G$	$f(x) = a^x$ $f(x + r) = f(x)$
Discrete logarithm	$\mathbf{Z}_r \times \mathbf{Z}_r$ + (mod r)	$\{a^j\}$ $j \in Z_r$ $a^r = 1$	$(\ell, -\ell s)$ $\ell, s \in \mathbf{Z}_r$	$f(x_1, x_2) = a^{kx_1 + x_2}$ $f(x_1 + \ell, x_2 - \ell s) = f(x_1, x_2)$
Order of a permutation	$\mathbf{Z}_{2^m} \times \mathbf{Z}_{2^n}$ + (mod 2^m)	$\mathbf{Z}_{2^n}$	$\{0, r, 2r, \ldots\}$ $r \in X$	$f(x, y) = \pi^x(y)$ $f(x + r, y) = f(x, y)$ π = permutation on X
Hidden linear function	$\mathbf{Z} \times \mathbf{Z}$, +	$\mathbf{Z}_N$	$(\ell, -\ell s)$ $\ell, s \in X$	$f(x_1, x_2) =$ $\pi(sx_1 + x_2 \bmod N)$ π = permutation on X
Abelian stabilizer	(H, X) H = any Abelian group	any finite set	$\{s \in H \mid$ $f(s, x) = x,$ $\forall x \in X\}$	$f(gh, x) = f(g, f(h, x))$ $f(gs, x) = f(g, x)$

Figure 1: Specific examples of the HSP. Credits: Nielsen and Chuang [1].

The General HSP Algorithm

(1) Prepare two registers in the state $|0\rangle|0\rangle$, each large enough to store an element of the group under consideration.

(2) Apply the quantum Fourier transform (QFT) on the first register.

(3) Store the result of applying the function f of the particular problem under consideration on the first register in the second register (this will be typically achieved by some sort of a controlled-f gate).

(4) Measure the second register.

(5) Apply the QFT on the first register and measure. This will yield an element of the dual $H^{\perp}$ of the hidden subgroup H.

(6) Run multiple simulations to find n elements of $H^{\perp}$. This will yield n linear equations relating the n elements of H. Solving them tells us what H is.

3.2.1 Solving Simon's Problem using HSP algorithm

Now we will see how the recipe given above can help us in solving Simon's problem. In this problem, we are given a function $f : \{0,1\}^n \to X$, X being any finite set, such that there is an $a \in \{0,1\}^n$ with $a \neq 0^n$ and

- $\forall x \quad f(x \oplus a) = f(x)$.
- if $f(x) = f(y)$ then either $x = y$ or $y = x \oplus a$. In the first case, the function is one-to-one, and in the second case, it is two-to-one.

We want to find a, i.e. determine whether the function is one-to-one or two-to-one. so we employ the HSP algorithm. The first stage allows us to set up a random coset state a:

$$\phi = 1/\sqrt{2}|z\rangle + 1/\sqrt{2}|z + a\rangle \tag{3}$$

where z is a random n bit string. We obtain the state of the first register as a superposition over exactly those values of x that are consistent with those contents for the second register. Hence, when we observe first register after doing a Fourier transform, we'll see a y such that $y \cdot a = 0$. Therefore, the the measurement output is a random y such that $y \cdot a = 0$. Furthermore, each y such that $y \cdot a = 0$ has an equal probability of occurring. We can obtain the following equation constraints on our system:

$$y_1 a_1 \oplus \cdots \oplus y_n a_n = 0.$$

In this case the group G is defined over the vector space Z_2^n and the hidden subgroup H is $0, a$. We obtain a random $k \in Z_2^n$ such that $k \cdot s = 0$ by doing Fourier sampling such that repeating this $n-1$ times will give us $n-1$ linear constraints as we mentioned in the preceding paragraph.

In order to implement on the Simon's algorithm, or any other specific algorithm, the working of the oracle or query function has to be understood; the oracle f_b does the searching for a hidden bitstring. Given a bitstring, $f_b(x) = f_b(y)$ if and only if $y = x \oplus b$; if the bitstring is all-zero, f_b is one-to-one function, while a bitstring that is not all-zero means f_b is two-to-one [4].

The state $|x\rangle |0\rangle$ is input to the oracle (when it's two-to-one). The value of the secret bit string $b \in \{0,1\}^n$ determines how the oracle changes the input second register. For all $x \in 0, 1^n$, the output is $|x\rangle |f(x \oplus a)\rangle$. For details as to how to implement this in an algorithm, see the Qiskit textbook [4].

3.3 Simulations

The major theme explored in our project is that the HSP is the most general framework for analyzing and solving problems that involve finding the period of some function in some sense, and that many well-known quantum algorithms, such as the ones given in Fig 1, are just specific instances of the HSP. To simulate these specific problems on a quantum computer using the general algorithm we have developed above, all one has to do is to find a way to implement the function f efficiently, so that step 3 in the general algorithm can be concretely realised (all the steps are detailed in the Jupyter notebook). In this section, we will do precisely this. We will specialise the general algorithm to Simon's problem and order finding, and show the results of simulating them on an IBM computer and Qiskit aeres module, respectively. We will also compare our results with those of the traditional quantum algorithms for Simon's problem and period-finding available on the Qiskit website. We will find that we get the same results in either case. This would furnish concrete examples of the fact that the HSP is evidently a more general version of specific problems that involve periodic functions.

To implement the function specific to each problem, we follow the Qiskit textbook [4]. Further details regarding the Python codes for our algorithms can be found in the Jupyter notebook accompanying this report.

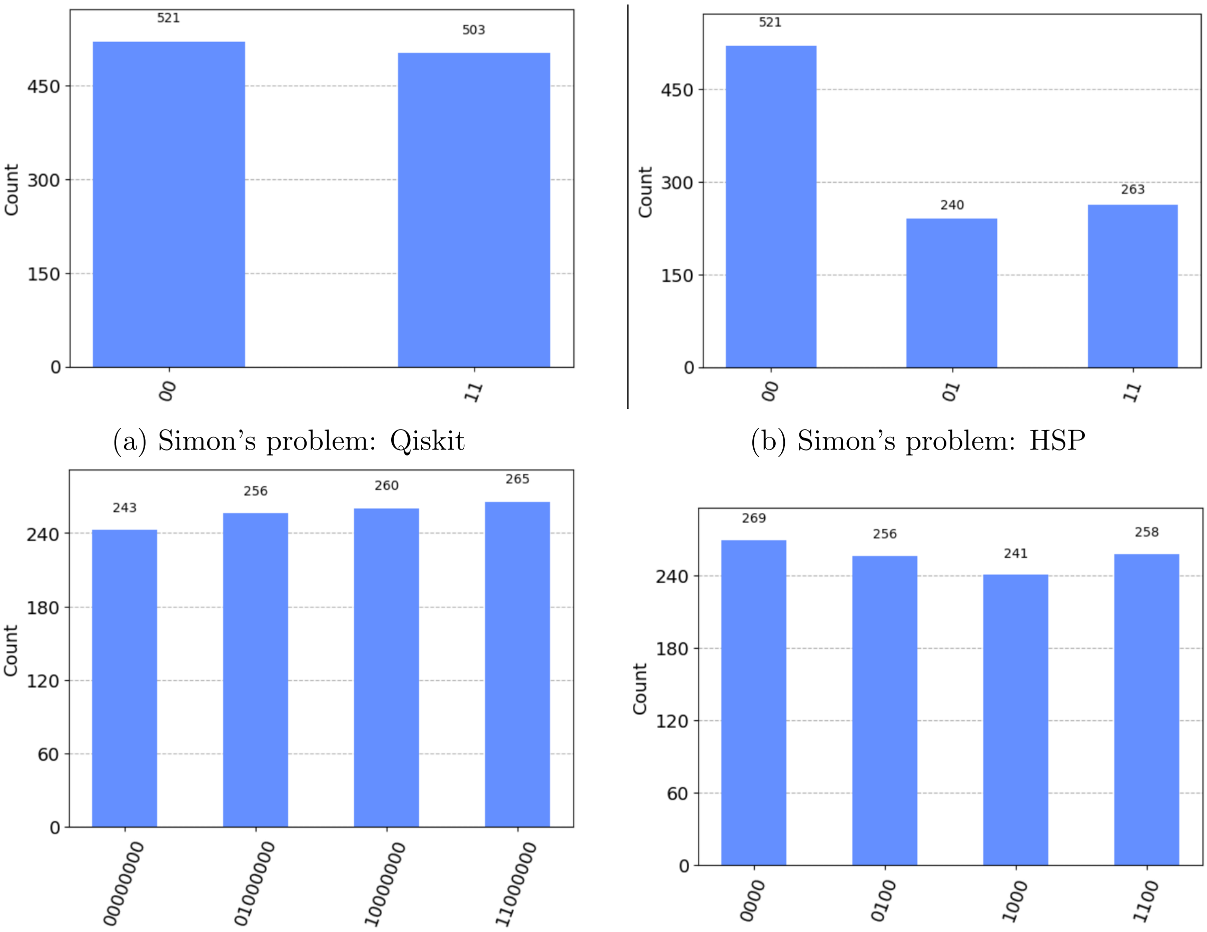
Fig 2 shows the results of simulations[1] for all algorithms. The algorithms for Simon's

[1] The results for both problems are those of simulations on classical computer, such as the ibm-qasm-simulator. For order finding, this was inevitable, since the IBM quantum computers that are available for free use allow at most 7 qubits, whereas order finding for even two-digit numbers requires at least 8. For Simon's problem, as we also mention in the accompanying Jupyter notebook, the codes we ran on a real quantum computer involved a wrong implementation of the function f, and when we realised our mistake, it was too late to run another simulation on a real quantum computer before the deadline of this project. Nevertheless, we have fixed our codes.

problem implemented the function f corresponding to a hidden bit string $b = 11$, while those for order finding were tailored to finding the order of 7 mod 15, which is 4 or equivalently 0100 in binary notation.

The HSP and Qiskit results for order finding agree, confirming our prediction. The discrepancy in the HSP and Qiskit results for Simon's problem can be traced back to the fact that for some reason, the implementation of the function f for the hidden bit string $b = 11$ actually yielded the bit string 01 when used in our HSP algorithm; we did the calculation by hand as well, to remove any doubts as to the correctness of the simulators used to run our codes. Whichever string one regards to be the correct string for which the HSP algorithm is supposed to work, the occurrence of the other string in the results can probably be attributed to noise, which in turn may be an artefact of the HSP algorithm using the quantum Fourier transform, as opposed to the Hadamard gates; the simulators used perhaps model the circuits differently, leading to the appearance of noise in one but not the other.

In any case, the agreement between the HSP and qiskit results for order finding does lend credence to our HSP algorithm. Indeed, looking at Figs. 5 and 6, we see that apart from the implementation of the function f via a series of controlled unitary phase gates in the middle, the HSP and Qiskit circuits are quite different. It is thus a nontrivial observation that both these circuits yield the same results upon simulations. Therefore, these considerations concretely vindicate the validity of the general HSP algorithm in solving all kinds of problems that hinge on the type of behaviour of a function on its domain that we have made precise in the formulation of the HSP. Of course, we had established this validity in a rigorous mathematical sense in Section 3.2, but it is instructive to see some concrete examples, as we have done in this section.



(a) Simon's problem: Qiskit

(b) Simon's problem: HSP

(c) Order finding: Qiskit

(d) Order finding: HSP

Figure 2: Results of simulations.

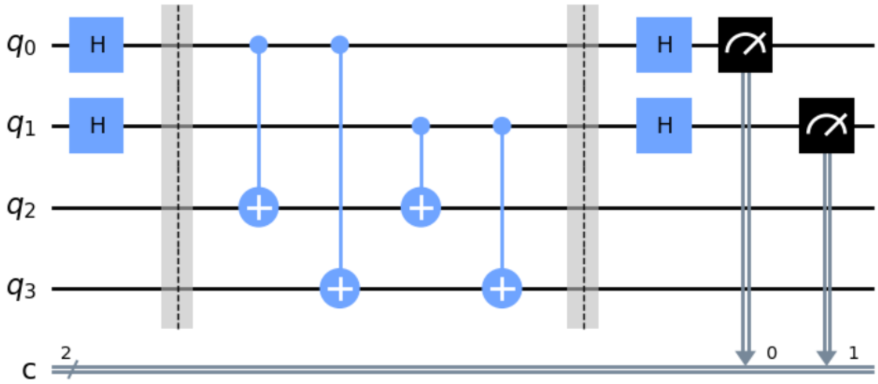


Figure 3: Circuit for Simon's algorithm: Qiskit version.

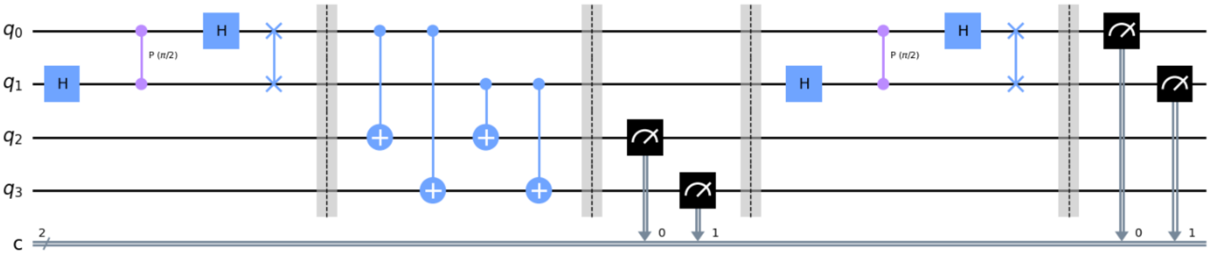


Figure 4: Circuit for Simon's algorithm: HSP version.

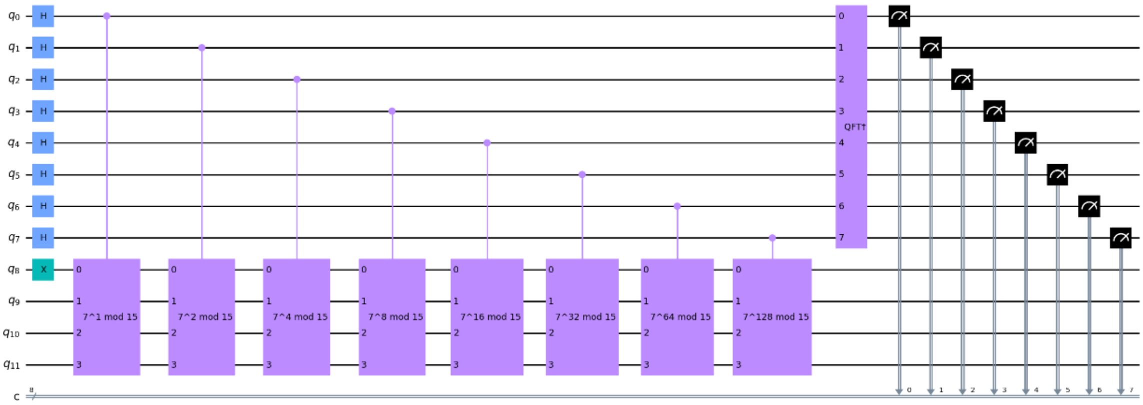


Figure 5: Circuit for order finding: Qiskit version.

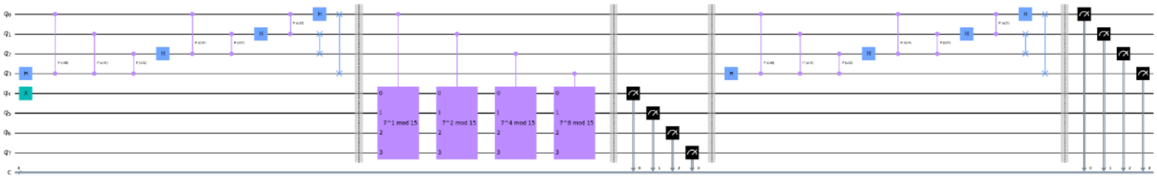


Figure 6: Circuit for order finding: HSP version.

4 Conclusion

In this report, we have explained the HSP, and clarified how it is a general framework for analysing problems that involve functions which are periodic in an appropriate sense, which can most easily described in terms of group theory. Based on our characterisation of HSP, we developed an algorithm to implement it, and subsequently applied it to Simnon's problem and order finding, comparing our results with the algorithms for the same problem found in the Qiskit textbook, and finding our results to be in close agreement, modulo the difficulty associated with the implementation of the function f in the case of Simon's problem. In view of this, one future avenue to explore is to precisely find out why the given implementation of this function seems to fail in the HSP case. Nevertheless, the theorems in Section 2 and the statement of the problem in Section 3.1 suffice to mathematically establish the validity of the claim that the HSP is indeed the most general framework for the kinds of problems under consideration.

References

[1] Nielsen, Michael A., and Isaac L. Chuang. Quantum Computation and Quantum Information. Cambridge: Cambridge University Press, 2022.

[2] Dummit, David Steven, and Richard M. Foote. Abstract Algebra. Danvers: John Wiley & Sons, 2004

[3] Vazirani, Umesh. Quantum Computation, Course Notes. Spring 2007. http://people.eecs.berkeley.edu/ vazirani/quantum.html.

[4] Qiskit Textbook. https://learn.qiskit.org/.