Quantum Computing and the Hunt for Hidden Symmetry

Gorjan Alagic* Alexander Russell†

1 Introduction

In 1994, Peter Shor gave efficient quantum algorithms for factoring integers and extracting discrete logarithms [20]. If we believe that nature will permit us to faithfully implement our current model of quantum computation, then these algorithms dramatically contradict the Strong Church-Turing thesis.[1] The effect is heightened by the fact that these algorithms solve computational problems with long histories of attention by the computational and mathematical communities alike.

In this article we discuss the branch of quantum algorithms research arising from attempts to generalize the core quantum algorithmic aspects of Shor's algorithms. Roughly, this can be viewed as the problem of generalizing algorithms of Simon [21] and Shor [20], which work over abelian groups, to general nonabelian groups.

The article is meant to be self-contained, assuming no knowledge of quantum computing or the representation theory of finite groups. We begin in earnest in Section 2, describing the problem of *symmetry finding*: given a function $f : G \to S$ on a group G, this is the problem of determining $\{g \in G \mid \forall x, f(x) = f(gx)\}$, the set of *symmetries* of f. We switch gears in Section 3, giving a short introduction to the circuit model of quantum computation. The connection between these two sections is eventually established in Section 4, where we discuss the representation theory of finite groups and the quantum Fourier transform - a unitary transformation specifically tuned to the symmetries of the underlying group. Section 4.2 is devoted to *Fourier*

*University of Connecticut, alagic@math.uconn.edu

†University of Connecticut, acr@cse.uconn.edu Supported by NSF CAREER award CCR-0093065, NSF grants EIA-0523456 and EIA-0523431, and ARO-ARDA grant 47976-PH-QC.

[1] By this we mean the statement "Any reasonable model of computation can be efficiently simulated on a probabilistic Turing machine." [5]

sampling, the basic algorithmic method that connects the symmetry finding problem and the Fourier "symmetry" basis effected by the quantum Fourier transform. Finally, we discuss some algorithmic successes in Section 5.

The reader should be cautioned that in many places we have forsaken precision for the sake of improved readability (such as it is), and have made no attempt to survey the rich and fascinating landscape of quantum algorithms. In fact, our selection of algorithms to highlight in Section 5 is motivated by an attempt to emphasize the relationship between representation theory and quantum algorithms; as such, many of the exciting technical advances in this area are unrepresented. The reader is encouraged to explore these other directions; indeed, one recent and closely related development is the discovery of efficient algorithms for finding hidden nonlinear structures in vector spaces over finite fields [6, 7].

2 Symmetries and Computation

Groups were invented to abstract the concept of symmetry. After all, if A is a geometric object then the identity map on A is surely a symmetry; furthermore, composing two symmetries ought to result in a symmetry, as should "inverting" a symmetry. The computational problem we describe below is the problem of *inferring* the family of symmetries of a given object (typically combinatorial rather than geometric, in our story). For example, the group S_{10} acts on the Petersen graph, of Figure 1, by permuting the 10 vertices. The symmetries of the Petersen graph under this action are those permutations that preserve incidence—these symmetries form a subgroup isomorphic to S_5. [2]

Let G be a finite group that *acts* on a set X. This means that we associate with every group element g a permutation π_g of the set X in such a way that the group operation of G is respected: $\pi_g \circ \pi_h = \pi_{gh}$. Thus the map $g \mapsto \pi_g$ is a *homomorphism* from G into the group of all possible permutations of X. We will write $g \cdot x$ or gx for $\pi_g(x)$ when it won't cause confusion. Examples of groups acting on sets are everywhere: *(i.)* The group S_n of all permutations of n letters acts on n letters by...permutation! *(ii.)* Any group G acts on itself by left-multiplication, associating with each element g the permutation $\pi_g : h \mapsto gh$. *(iii.)* The group A_4 acts as the symmetries of a regular tetrahedron. For more discussion, see Dummit and Foote's text [8]. Observe

[2] One pleasing way to view the S_5 action on the Petersen graph P is to identify the vertices of P with the 10 subsets of $\{1, \ldots, 5\}$ of size 2 in such a way that adjacent vertices correspond to pairs of subsets with nontrivial intersection; the obvious action of S_5 yields the entire family of automorphisms of P.

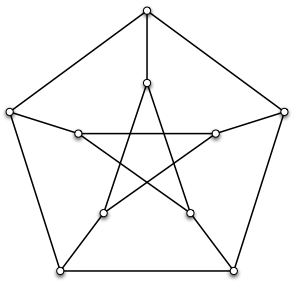

Figure 1: The Petersen graph.

that if G acts on the set X, then it also acts on the set of all *functions* $\{f : X \to S\}$ by the rule

$$g \cdot f : x \mapsto f(g^{-1} \cdot x)\,. \tag{1}$$

(Here $g \in G$, $f : X \to S$ is a function, and the $^{-1}$ is introduced so that this action composes correctly: $g_1(g_2(f)) = (g_1 g_2)f$.) For a function f on X, the *symmetries* of f are the group elements g for which $g \cdot f = f$. These elements form a subgroup $S(f) = S_G(f)$ of G. While we will often drop the subscript G in this notation, $S_G(f)$ does of course depend both on the choice of G and the details of the action of G on X.

Our signature computational problem shall be the problem of *determining the symmetries $S(f)$ of a function f*. We shall be very generous regarding the "presentation" of the function f (and the group G), merely asking that $f : X \to S$ be provided as an oracle. We shall assume, among other things, that the elements of X (and G) have a canonical representation as strings of length $O(\log |X|)$ (and $O(\log |G|)$) and that we can perform operations such as g^{-1}, $g_1 g_2$, and $g \cdot x$ in polynomial time in the lengths of these representations.

Before any discussion of quantum computation, we advertise below a number of interesting computational problems that directly reduce to symmetry finding.

2.1 Order finding

Let us consider the possible symmetries of a function $f : \mathbb{Z}_n \to S$, where $\mathbb{Z}_n = \{0, \ldots, n-1\}$ is the cyclic group of size n acting on itself by translation (i. e., addition modulo n). When is f invariant under a translation? This is the problem of *period finding*: we say that a function f on $\mathbb{Z}_n$ is *t-periodic* if $\forall x, f(x + t) = f(x)$. The least such integer t is called the *period* of f. It is easy to show that f is m-periodic if and only if m is an integer multiple

of the period t. Evidently, the subgroup $S(f)$ of all symmetries of f is the cyclic subgroup of $\mathbb{Z}_n$ generated by t. In this case, the group in question is $G = \mathbb{Z}_n$, and the set on which the group acts is $X = \mathbb{Z}_n$ as well.

A similar problem, that of *order finding*, is a central component of Shor's celebrated quantum algorithm for factoring [20]. Recall that for a prime p, the group $\mathbb{Z}_p^*$ is the set $\{1, \ldots, p-1\}$ under multiplication modulo p. Order finding is the problem of determining the (multiplicative) order of an element x of $\mathbb{Z}_p^*$, i. e., determining the least integer t such that $x^t \equiv 1 \bmod p$. If we define $f : \mathbb{Z}_{p-1} \to \mathbb{Z}_p^*$ by the rule

$$f : s \longmapsto x^s \bmod p$$

then the period of the function f is one more than the order of x. Thus order finding reduces to symmetry finding. In order finding, the group $\mathbb{Z}_{p-1}$ acts on itself, so that $G = X = \mathbb{Z}_{p-1}$.

2.2 Hidden shifts of the Legendre symbol

Let p be a prime number. A nonzero element x of $\mathbb{Z}_p$ is called a *quadratic residue* modulo p if there exists an integer n such that $n^2 \equiv x \bmod p$. It is not hard to check that setting

$$\chi_2 : x \longmapsto \begin{cases} 0 & \text{if } x = 0; \\ 1 & \text{if } x \text{ is a quadratic residue modulo } p; \\ -1 & \text{otherwise.} \end{cases}$$

defines a multiplicative function on $\mathbb{Z}_p$ (so that $\chi_2(yz) = \chi_2(y)\chi_2(z)$). The image of x under χ_2 is known as the *Legendre symbol* of x modulo p. "Shifting" the function χ_2 has been proposed as a pseudorandom generator [22]: specifically, fixing a prime p and a length $\ell > \log p$, translation of χ_2 by a randomly selected element $t \in \mathbb{Z}_p$ defines a sequence

$$\chi_2(t), \chi_2(t+1), \ldots, \chi_2(t+\ell)\,.$$

One way to "break" such a generator is to completely recover t from this sequence of values. To place the problem of breaking the pseudorandom generator in our context, we consider the (potentially easier) problem of determining t given oracle access to the entire shifted function

$$f_t : x \longmapsto \chi_2(x+t)\,.$$

Though the sequence of bits $\chi_t(0), \ldots, \chi_t(\ell)$ has been conjectured to be pseudorandom for appropriate values of ℓ, the function f_t defined above does

possess some rich symmetries. To capture these symmetries, we introduce the group of *affine linear transformations* of $\mathbb{Z}_p$. A function $\alpha : \mathbb{Z}_p \to \mathbb{Z}_p$ is *affine linear* if it has the familiar form $\alpha(x) = ax + b$, where $a \neq 0$. (Here $a, b \in \mathbb{Z}_p$ and all arithmetic is done modulo p.) These functions form a nonabelian group, denoted $\mathbb{A}_p$, under composition: observe that if $\alpha(x) = ax + b$ and $\alpha'(x) = a'x + b'$ then $\alpha \circ \alpha'(x) = aa'x + ab' + b$.

Recalling that $\mathbb{A}_p$ acts on the *functions on* $\mathbb{Z}_p$ via equation (1), we investigate the symmetries of f_t under $\mathbb{A}_p$. A short calculation shows that if $\alpha(x) = ax + t(a-1)$ and, furthermore, $\chi_2(a) = 1$, then

$$\begin{aligned}[\alpha^{-1} \cdot f_t](y) &= f_t(ay + t(a-1)) = \chi_2(ay + t(a-1) + t) \\ &= \chi_2(a(y+t)) = \chi_2(a)\chi_2(y+t) \\ &= \chi_2(y+t) = f_t(y)\,.\end{aligned}$$

It is easy to check that elements of the form $x \mapsto ax + t(a-1)$, where a is a quadratic residue, each represent a symmetry of f; together these form $S(f_t)$, the symmetry subgroup of f_t (under this $\mathbb{A}_p$ action). Moreover, each f_t induces a *different* subgroup of $\mathbb{A}_p$; evidently, solving this "hidden shift" problem can be reduced to symmetry finding under the group $\mathbb{A}_p$, acting on the set $X = \mathbb{Z}_p$.

2.3 Graph automorphism and isomorphism

Let $\mathcal{G} = (V, E)$ denote a simple undirected graph with vertex set $V = \{1, \ldots, n\}$ and edge set E. An *automorphism* of $\mathcal{G}$ is a permutation π of the vertices V that preserves incidence: $(v, w) \in E \Leftrightarrow (\pi(v), \pi(w)) \in E$. These automorphisms, taken together, form a subgoup $\mathrm{Aut}(\mathcal{G})$ of S_n, the group of all permutations of $V = \{1, \ldots, n\}$. The *graph automorphism* problem is the problem of determining if $\mathrm{Aut}(\mathcal{G})$ is nontrivial (that is, if there is a nontrivial automorphism of $\mathcal{G}$).

Let us express this problem in our framework of symmetry finding. First, we let a permutation $\pi \in S_n$ act on an edge (u, v) by the rule $\pi(u, v) = (\pi(u), \pi(v))$; it acts on the edge set E by $\pi E = \{\pi(u, v) \mid (u, v) \in E\}$. Define the function $s_{\mathcal{G}}$ on S_n by the rule

$$s_{\mathcal{G}}(\pi) = \pi E\,.$$

Then $\mathrm{Aut}(\mathcal{G})$ is precisely the group $S(s_{\mathcal{G}})$: thus solving the symmetry finding problem in this case determines the entire automorphism group of $\mathcal{G}$ (and, in particular, the answer to the graph automorphism problem for this graph). This is another example of a symmetry finding problem, where the group is $G = S_n$ acting on itself.

We remark that the task of determining if two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are *isomorphic* reduces to the above situation by considering the automorphisms of the (disjoint) union $\mathcal{G}_1 \cup \mathcal{G}_2$. Indeed, if the graphs are not isomorphic, the only automorphisms of $\mathcal{G}_1 \cup \mathcal{G}_2$ would be of the form $\pi \cup \sigma$, where π is an automorphism of $\mathcal{G}_1$ and σ is an automorphism of $\mathcal{G}_2$. On the other hand, if $\tau : \mathcal{G}_1 \to \mathcal{G}_2$ is an isomorphism, then there exists an automorphism of $\mathcal{G}_1 \cup \mathcal{G}_2$ which swaps $\mathcal{G}_1$ and $\mathcal{G}_2$, and then applies $\tau^{-1} \cup \tau$.

3 Quantum computing

For the sake of analogy, we begin by retelling the story of classical computation: beginning with an input vector $\mathbf{v} \in \{0,1\}^n$, we carry out a sequence of *local transformations*, each of which affects but a constant number of coordinates by subjecting them to some function $t_i : \{0,1\}^c \to \{0,1\}^c$. In order for this story to really capture the familiar circuit model, we need to be more generous with "workspace," initially embedding $\mathbf{v}$ into the first n coordinates of $\{0,1\}^\omega$. Finally, we honor the first bit of the result by allowing it to determine the *outcome* of this computation which, in this way, determines a Boolean function. The story is most exciting when it realizes some Boolean function we care about as a particularly short (or particularly nicely-structured) sequence of transformations.

The story of quantum computation can be told by "unitarily extending" the tale above: beginning with an input vector $\mathbf{v} \in H^{\otimes n}$, we carry out a sequence of *local unitary transformations*, each of which affects but a constant number of coordinates by subjecting them to an arbitrary unitary (i. e., length-preserving linear) operator $t_i : H^{\otimes c} \to H^{\otimes c}$. The first unfamiliar character appearing in the quantum tale is H, which we take to be a 2-dimensional complex vector space, on which we have an inner product $\langle \cdot, \cdot \rangle$ and hence also a notion of length: $\|x\|^2 = |\langle x, x \rangle|$. To maximize our analogy with the classical story, we select an orthonormal basis for H consisting of two orthonormal vectors named $|0\rangle$ and $|1\rangle$. Leaving the details of what we mean by $H^{\otimes n}$ and a unitary operator "affecting but a constant number of coordinates" to the reader's imagination for the moment, we continue to say that in this quantum case, as well, we shall embed $H^{\otimes n}$ inside $H^{\otimes \omega}$ to be equally generous regarding workspace and, furthermore, allow the first coordinate of the resulting state to determine the outcome of this computation in an appropriate fashion. For our purposes, an *efficient* quantum computation is one that involves a polynomial number of local transformations.

Despite whatever "compelling analogies" the reader has been cajoled to accept by the story above, this computational model appears complicated

and suspicious, especially if you have been brought up on a healthy diet of recursive function theory, Turing machines, and classical circuit complexity. However, this model of "unitary evolution" is one of the essential features of quantum mechanics, a physical model that predicts both qualitative and quantitative features of the small-scale behavior of the universe. To the best of our current knowledge, the many fundamental curiosities and surprises that manifest in quantum mechanics are facts of nature, though they be wildly inconsistent with our physical intuition born of a lifetime of experience with classical macroscopic phenomena. Engineering advances of the last decade demonstrate that these quantum phenomena are available for human tinkering and, hopefully, for the highly-structured sort of tinkering that would comprise "computation."

Let us return to flush out the remaining details of the model of quantum computation introduced above. The state of the computation, throughout the process above, lies in $H^{\otimes n}$, the n-fold *tensor* power of H. In general, if A and B are vector spaces with orthonormal bases $\{\mathbf{a}_i\}$ and $\{\mathbf{b}_j\}$, a *tensor product* of these spaces is a vector space, denoted $A \otimes B$, along with a map $(\mathbf{a}, \mathbf{b}) \mapsto \mathbf{a} \circ \mathbf{b}$ of $A \times B$ to $A \otimes B$ so that $\circ$ is linear in each coordinate and, furthermore, the set $\{\mathbf{a}_i \circ \mathbf{b}_j\}$ is a basis for $A \otimes B$. As we assume that A and B have an inner product, we can naturally define an inner product on $A \otimes B$ by extending the rule $\langle \mathbf{a} \circ \mathbf{b}, \mathbf{a}' \circ \mathbf{b}' \rangle = \langle \mathbf{a}, \mathbf{a}' \rangle \langle \mathbf{b}, \mathbf{b}' \rangle$; this will, in particular, make the basis $\mathbf{a}_i \circ \mathbf{b}_j$ above orthonormal. Unfolding the definition and trusting that $\circ$ is associative (it is), we find that $H^{\otimes n}$ is a vector space with 2^n orthonormal basis vectors

$$|a_1\rangle \circ \cdots \circ |a_n\rangle \ , \quad (a_1, \ldots, a_n) \in \{0,1\}^n \, ,$$

one for each binary string in $\{0,1\}^n$! It is customary to use the symbol $\otimes$ for the map $\circ$ above (even though this overloads the symbol, which we also used to denote the vector space $A \otimes B$); furthermore, it is customary in this case to use the notation $|a_1 \ldots a_n\rangle$ for the vector $|a_1\rangle \otimes \cdots \otimes |a_n\rangle$.

This apparent relationship between the basis of $H^{\otimes n}$ and the "intermediate states" that appeared in our notion of *classical* computation is not an accident. Indeed, our rule for feeding (classical) inputs to a quantum machine will be to identify the classical input $a \in \{0,1\}^n$ with the unit-length basis vector $|a\rangle \in H^{\otimes n}$.

It is easy to check that if U is a unitary operator on the vector space A, then U naturally defines a unitary operator on $A \otimes B$ by the rule $\mathbf{a} \otimes \mathbf{b} \mapsto (U\mathbf{a}) \otimes \mathbf{b}$. It is precisely this sense in which the local operators discussed in the definition of computation above are applied to a "constant number of indices." The sequence of unitary (and hence length-preserving) operators associated

with a quantum computation thus carries any unit length input vector $\mathbf{a}$ to a unit length result $\mathbf{r}$. While we naturally have a clear interpretation of the input vector $\mathbf{a}$ as an element of $\{0,1\}^n$, the result $\mathbf{r}$ is typically a linear combination

$$\mathbf{r} = \sum_{a \in \{0,1\}^n} \alpha_a |a\rangle \ , \quad \alpha_a \in \mathbb{C}$$

with exponential support. What we *can* be sure of, however, is that $\mathbf{r}$ has unit length: $\sum_a |\alpha_a|^2 = 1$. These amplitudes $|\alpha_a|^2$ evidently give rise to a probability distribution on $\{0,1\}^n$ - this is how we shall extract a classical (though stochastic) output from a quantum computation; indeed, we shall say that the probability that the computation *accepts* is the probability that the first bit is a 1 according to this probability distribution.

We remark that this curious method for "terminating" our quantum computation is, along with the rule of unitary evolution, directly borrowed from the model of quantum mechanics used to model the universe: this is the process of *measurement*. This is the second time we have justified a potentially surprising aspect of the model by direct appeal to the mathematical foundations of quantum mechanics. Indeed, one of the original motivations for defining (and yearning for) such a model is that it would allow us to "simulate" and study quantum systems of physical interest. Especially after hearing this explanation for the genesis of the model, it is natural to wonder if the new features of the model offer any new traction in the *classical* computational arena. We'll see below is that the answer is yes: quantum computation can, in some cases, uncover exactly the combinatorial symmetries mentioned in Section 2.

We describe, in the next section, how these symmetries are related to the unitary evolution (and measurement) in the model above. To briefly advertise what comes ahead, however, we remark that with any finite group G one may inexorably associate a finite collection of "harmonic objects," objects that precisely capture the symmetry structure of G. These objects, taken together, define a unitary basis change in the set of $\mathbb{C}$-valued functions on G. For many groups of interest, this unitary basis change can be efficiently carried out on a quantum computer, thus exposing the symmetries of objects on which G acts.

4 Group harmonics and applications

4.1 Representations of finite groups

Let G be a group acting on a set X; as in Section 2, we may "extend" this group action to the set of functions $\mathbb{C}X \triangleq \{f : X \to \mathbb{C}\}$ by the rule

$$(\pi f)(x) = f(\pi^{-1}x)\,.$$

This case, where we consider $\mathbb{C}$-*valued functions* on X (rather than the various "combinatorial" choices taken in Section 2) is especially attractive because we can bring to bear the tools of linear algebra to study these actions and their symmetries. To emphasize this connection, we observe that $\mathbb{C}X$ is a complex vector space with a distinguished basis: the delta functions $f_x : y \mapsto \delta_{xy}$ (where δ_{xy} is equal to one when $x = y$ and is zero otherwise). Notationally, if we let $|x\rangle$ denote the function f_x above, elements of $\mathbb{C}X$ can be written

$$\sum_{x\in X} a_x\,|x\rangle\ , \quad a_x \in \mathbb{C}\,,$$

a convention we shall adopt throughout. Now we can view the action of G on $\mathbb{C}X$ as a family of linear operators; in particular, each g is associated with a linear operator $\rho_g : \mathbb{C}X \to \mathbb{C}X$ in such a way that $\rho_g\rho_h = \rho_{gh}$.

Note that with this linear algebraic view of group actions, $g \in S(f)$ (that is, the group element g is a symmetry of a function f) *exactly when* f *is an eigenvector with eigenvalue* 1 of the map ρ_g. This suggests a general study of the eigenvalues and invariant spaces of these group actions, a lifestyle known as the *representation theory* of finite groups. The general definition is the following:

Definition 1. *Let G be a finite group. A* ***representation*** *of G is a homomorphism $\rho : G \to \mathsf{GL}(V)$, where $\mathsf{GL}(V)$ is the collection of invertible linear operators on a (finite dimensional) $\mathbb{C}$-vector space V. The* ***dimension*** *of ρ, denoted d_ρ, is the dimension of V.*

Observe that ρ assigns to each group element g a linear operator $\rho(g)$ so that $\rho(g)\rho(h) = \rho(gh)$, just as in our permutation example above. Indeed, any action of G on a set X immediately induces a representation of G (on $\mathbb{C}X$). The representations of a group G offer a principled approach to the problem of understanding symmetries under G and its subgroups. We remark that when G is finite (as always, in this article), we may assume without loss of generality that representations come equipped with an inner product for which each $\rho(g)$ is unitary.

What renders the theory especially attractive is that a given finite group G possesses a finite number of atomic "irreducible" representations, in terms of which all others can be expressed. To develop this decomposition machinery, we set down a few more definitions. We say that two representations $\rho_1 : G \to GL(V_1)$ and $\rho_2 : G \to GL(V_2)$ are *equivalent* when they are related by a (linear) isomorphism $E : V_1 \to V_2$; specifically, for every g we have $\rho_2(g) = E\rho_1(g)E^{-1}$. (If these ρ_i happened to be defined on the same space, they would be related by a change of basis.) The familiar notion of direct sum can be applied to sew together a pair of representations: if $\rho_1 : G \to GL(V_1)$ and $\rho_2 : G \to GL(V_2)$ are two representations of G, we may construct a new representation on $V_1 \oplus V_2$ with the action

$$\mathbf{v} \oplus \mathbf{w} \mapsto \rho_1(g)\mathbf{v} \oplus \rho_2(g)\mathbf{w} \,;$$

we name this representation $\rho_1 \oplus \rho_2 : G \to GL(V_1 \oplus V_2)$. In matrix form, $\rho_1 \oplus \rho_2(g)$ can be realized as a block diagonal matrix, with each of the $\rho_i(g)$ forming one of two blocks.

Finally returning to the notion of irreducibility promised above, if $\rho : G \to GL(V)$ is a representation, we say that a subspace W of V is *invariant* when each $\rho(g)$ fixes W as a space. Of course, V and $\{0\}$ are always invariant. When these are the *only* invariant spaces, the representation is said to be *irreducible*. As mentioned above, a finite group G has a finite number of distinct irreducible representations up to equivalence; we let $\hat{G}$ denote this finite set of (equivalence classes of) irreducible representations. Every other representation of G can expressed as a direct sum of copies of representations in $\hat{G}$.

Our discussion of group actions in Section 2 leads us to naturally define another representation. Recall that any group G acts on itself by left-multiplication. As before, this allows G to act on the space $\mathbb{C}G = \{f : G \to \mathbb{C}\} = \text{span}\,(\{|g\rangle \mid g \in G\})$ via

$$g : |h\rangle \mapsto |gh\rangle \ .$$

The representation so defined is called the *left regular representation* of G. We remark that this representation is not irreducible (unless $|G| = 1$). In particular, note that the element $\sum_{g\in G} |g\rangle$ is fixed by *any* permutation, and in particular by those that correspond to left multiplication by elements of G. The linear span of this element is clearly a one-dimensional invariant subspace of $\mathbb{C}G$. This same argument applies to any permutation representation: evidently, every representation discussed in the article thus far is reducible!

We remark that if the full symmetric group S_n acts on

$$\mathbb{C}\{1,\ldots,n\} = \left\{ \sum_{i=1}^{n} a_i \left|i\right\rangle \mid a_i \in \mathbb{C} \right\}$$

by permuting the $\left|i\right\rangle$, the vector space perpendicular to the vector $\sum_i \left|i\right\rangle$ is an irreducible representation of dimension $n-1$. (The inner product used to make sense of the word perpendicular is the one consistent with the $\left|i\right\rangle$ basis: $\left\langle \left|i\right\rangle, \left|j\right\rangle \right\rangle = \delta_{ij}$.)

This example above might suggest that the general problem of decomposing a representation into irreducible pieces (or, indeed, even identifying irreducible representations) is quite difficult and, potentially, delicate. This is true; however, there is a remarkable tool that considerably simplifies these questions. Given an irreducible ρ, its *character* at a group element g is defined to be $\chi_\rho(g) \triangleq \mathbf{tr}\, \rho(g)$, the trace of $\rho(g)$. The important feature of these characters is that under the pairing

$$(\chi, \psi) = \frac{1}{|G|} \sum_g \chi(g)\psi(g^{-1}),$$

we find that for irreducible representations ρ and σ,

$$(\chi_\rho, \chi_\sigma) = \begin{cases} 1 & \text{if } \rho \text{ and } \sigma \text{ are equivalent,} \\ 0 & \text{otherwise.} \end{cases}$$

As it is clear that for two representations β and γ, $\chi_{\beta\oplus\gamma}(g) = \chi_\beta(g) + \chi_\gamma(g)$, this pairing offers an immediate method for determining if a given representations is reducible and, moreover, decomposing a representation into irreducibles. Indeed, observe that if β is a representation and σ irreducible then $(\chi_\beta, \chi_\sigma)$ is precisely the number of times σ appears in the decomposition of β.

With these tools it is possible to show that the left regular representation $R : G \to GL(\mathbb{C}G)$ of a group G has a *generic* decomposition into irreducible representations. In particular, every irreducible representation ρ of G appears in R exactly d_ρ times. We write

$$R = \bigoplus_{\rho\in\hat{G}} \rho^{\oplus d_\rho} = \bigoplus_{\rho\in\hat{G}} \underbrace{\rho \oplus \cdots \oplus \rho}_{d_\rho}. \tag{2}$$

In particular, counting dimensions on both sides of this equation yields the equality $|G| = \sum_\rho d_\rho^2$.

The Fourier transform The definition of $\mathbb{C}G$ immediately distinguishes the *group basis* of $\mathbb{C}G$: the elements $\{|g\rangle\}$. On the other hand, the direct sum decomposition of Equation (2) above provides an alternate (and, in general, transverse) means of describing elements of $\mathbb{C}G$: in terms of their projections into the spaces $\rho^{\oplus d_\rho}$. If $f : G \to \mathbb{C}$ is a function and $\rho \in \hat{G}$, the *Fourier transform of f at ρ* is the linear operator

$$\hat{f}(\rho) = \sqrt{\frac{d_\rho}{|G|}} \sum_{g \in G} f(g)\rho(g^{-1}) \,.$$

If we select a basis for the space V_ρ on which ρ operates, $\hat{f}(\rho)$ is realized as a $d_\rho \times d_\rho$ matrix of complex numbers. Taken over all $\rho \in \hat{G}$, the linear operators $\hat{f}(\rho)$ determine $\sum_\rho d_\rho^2 = |G|$ complex numbers—exactly the dimension of $\mathbb{C}G$. With the scaling factor $\sqrt{d_\rho/|G|}$ appearing above, this transformation from $f \in \mathbb{C}G$ to these matrices is unitary and actually corresponds to writing f in a basis consistent with the decomposition (2) of $\mathbb{C}G$ above. More prosaically, the Fourier transform is a unitary map from the group basis $\{|g\rangle\}$ to the basis

$$\{|\rho, i, j\rangle : \rho \in \hat{G} \text{ and } 1 \leq i, j \leq d_\rho\} \,, \tag{3}$$

where the i and j entries correspond to the rows and columns of the matrices $\hat{f}(\rho)$ above.

The reason for this long digression, and the critical fact that weds quantum computing and representation theory, is that the Fourier transform, as described above, can be efficiently computed on a quantum computer for many groups of interest [4, 15]. In this context, the unitary Fourier transform is called the *quantum Fourier transform*, and is a key ingredient in the algorithms we discuss in the sequel.

4.2 Looking for hidden symmetries in coset states

We now discuss a special case of the symmetry finding problem presented in Section 2. Suppose we are given oracle access to a function $f : G \to S$ and a promise that f is a *transversal* of some unknown subgroup H of G in the sense that

$$f(hx) = f(x) \Leftrightarrow h \in H \,.$$

For concreteness, we will assume that S is the set of integers $\{1, 2, \ldots, n\}$ for a suitably large n. As f is constant and distinct on each coset of H, the problem of determining $S(f)$ is precisely the problem of determining the "hidden" subgroup H. This problem is aptly titled the *Hidden Subgroup*

Problem (HSP). The standard method of *Fourier sampling* [5] for solving such problems on a quantum computer proceeds as follows.

Our first step is to prepare the "uniform superposition over G," i. e., the state

$$\frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle \ ,$$

and then "evaluate" the oracle function f on this state.[3] The resulting state of the system is

$$\frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle |f(g)\rangle \ . \tag{4}$$

Now, we wish to "measure the second register" (that is, the one containing $|f(g)\rangle$), thereby isolating the elements of G where f takes a particular value $k \in S$. In general, a quantum measurement on a vector space V is described by an orthogonal decomposition $V = W_1 \oplus \ldots \oplus W_k$; when such a measurement is applied to a vector $\mathbf{v}$, it results in the measured value i with probability $\|\Pi_i(\mathbf{v})\|^2$, where Π_i projects onto the subspace W_i. Since we are supposing that $\mathbf{v}$ has length 1, this yields a probability distribution on i. The state of our system after the measurement will be the (renormalized) projection

$$\frac{\Pi_i \mathbf{v}}{\|\Pi_i \mathbf{v}\|_2^2} \in W_i \ .$$

In our case, the state (4) lives in the space $V = \mathbb{C}G \otimes \mathbb{C}S$. We have specific bases in mind, too: the basis for the space $\mathbb{C}G$ is $\{|g\rangle : g \in G\}$, while the basis for $\mathbb{C}S$ is $\{|1\rangle, |2\rangle, \ldots, |n\rangle\}$. Our desired measurement corresponds to the orthogonal decomposition

$$V = [\mathbb{C}G \otimes |1\rangle] \oplus [\mathbb{C}G \otimes |2\rangle] \oplus \cdots \oplus [\mathbb{C}G \otimes |n\rangle] \ .$$

The result of this measurement will be the (renormalized) projection of the state (4) to one of the subspaces $\mathbb{C}G \otimes |k\rangle$; that is, it will be a uniform superposition over all elements of G where f takes the particular value k. The set of such group elements is, by the conditions we imposed on f, a coset of some hidden subgroup H. We are thus left in the "coset state"

$$\frac{1}{\sqrt{|H|}} \sum_{h \in H} |ch\rangle \ , \tag{5}$$

[3] To make this a unitary operation, we actually need some additional "empty" workspace. The quantum oracle is the map $U : |x, y\rangle \mapsto |x, y \oplus f(x)\rangle$, where $\oplus$ denotes the (clearly unitary) operation of bitwise addition. In our case, $x = \sum_{g \in G} |g\rangle$ and $y = |00 \cdots 0\rangle$, where the register containing y is large enough to accommodate the possible values of f.

where c is an unknown but, fortunately, random element of G. Next, we apply the Quantum Fourier Transform from Section 4.1. Let us first view the above state as the scaled characteristic function

$$\psi(g) = \begin{cases} 1/\sqrt{|H|} & \text{if } g \in cH, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Recall that the Fourier transform of this function, at an irreducible $\rho \in \hat{G}$, is a $d_\rho \times d_\rho$ matrix denoted by $\hat{\psi}(\rho)$. Thus, the Quantum Fourier Transform of the above state will be indexed by $|\hat{G}|$-many representations ρ, and d_ρ^2-many entries for each such ρ. Concretely, the new state is given by

$$\sum_{\rho,i,j} \hat{\psi}(\rho)_{ij} |\rho, i, j\rangle \ . \tag{7}$$

We think of this state as being indexed by three registers: the first register specifies the representation ρ, the second register specifies the "row" i, and the third register specifies the "column" j. Our task now is to perform some kind of measurement on the above state, and attempt to determine the hidden subgroup H based on this measurement. For instance, we may simply measure the first register - resulting in a particular representation σ with probability

$$\sum_{i,j} \left|\hat{\psi}(\sigma)_{ij}\right|^2 = \left\|\hat{\psi}(\sigma)\right\|_2^2 .$$

This method is called *weak Fourier sampling*. Alternatively, we may apply *strong Fourier sampling*, which will completely measure all of the registers. The result of strong sampling is a matrix entry $\hat{\psi}(\tau)_{kl}$, occurring with probability $|\hat{\psi}(\tau)_{kl}|^2$. (Note that this measurement depends on the basis in which σ is expressed by our Fourier transform.)

In the coming sections, we will describe how one can efficiently reconstruct normal subgroups via the weak Fourier sampling method discussed above. As any subgroup of an abelian group is normal, this will allow us to completely solve the HSP on abelian groups. In particular, this will imply that the period finding and order finding applications discussed in Section 2 have efficient solutions on a quantum computer.

Some progress has also been made in the arena of non-normal subgroups. For instance, Hallgren, Ip, and Van Dam [22] constructed an efficient quantum algorithm for the hidden shifts of the quadratic character discussed in Section 2 (their algorithm actually departs from the Fourier sampling framework above, directly using the oracle as "phase information"; the affine group

formulation of the problem presented in Section 2 was solved by strong sampling in [16]). Kuperberg [14] devised a "sieve" algorithm to solve the hidden subgroup problem in the dihedral groups D_n with subexponential running time $2^{O(\sqrt{n})}$. (We remark that no classical algorithm can have running time $2^{o(n)}$.) We will briefly discuss these algorithms in Section 5.2.2. Finally, in Section 5.3, we will discuss an approach developed by Bacon, Childs, and van Dam [3] that has given rise to efficient algorithms for various semi-direct product groups, including the Heisenberg groups $\mathbb{Z}_p \ltimes \mathbb{Z}_p^2$.

Sadly, our final example of Graph Automorphism, where we seek to find certain hidden subgroups of the symmetric group S_n, still defies the community's attempts at devising efficient algorithms. In fact, several negative results have shown that solving the HSP using coset states in the symmetric groups (and some other group families) *requires* so-called *multiregister* Fourier sampling. This entails sampling several coset states $\psi_1, \psi_2, \ldots, \psi_k$ and then performing some kind of (non-separable) measurement on the tensor product $\psi_1 \otimes \psi_2 \otimes \cdots \otimes \psi_k$ of the k registers. In the case of the symmetric group, we must use at least $\Omega(\log |G|)$ many samples [10, 18]. As we will later discuss, however, all is not lost even in this area. Indeed, even for some group families to which these negative results apply, there are quantum algorithms which perform better than any possible classical algorithm.

5 Algorithmic progress

5.1 Reconstructing normal subgroups

In this section, we will show how to resolve the HSP efficiently in the case where the hidden subgroup is normal. The results of this section are due to Hallgren, Russell, and Ta-shma [11]. As every subgroup of an abelian group is normal, our discussion will include the HSP on abelian groups as a special case. Recall from Section 4.1 that the Quantum Fourier Transform is the basis change operator that maps the group basis to the Fourier basis; that is, it rewrites a function $\psi \in \mathbb{C}G$ (indexed by group elements) as another function $\hat{\psi}$ (indexed by representations, rows, and columns). As discussed in Section 4.2, performing "weak Fourier sampling" on $\hat{\psi}$ will then measure a particular representation $\rho \in \hat{G}$, while ignoring the rows and the columns. The probability of observing ρ is equal to the norm $\|\hat{\psi}(\rho)\|^2$ of the Fourier transform of ψ at ρ.

We first let ψ be the "coset state" (5) from Section 4.2; it's easy to show that our analysis will not depend on the value of c, and so we assume that $c = 1$. Then ψ takes the value $1/\sqrt{|H|}$ on the hidden subgroup H of G,

and is zero elsewhere. The Fourier transform of ψ at an irreducible ρ is then given by

$$\hat{\psi}(\rho) = \sqrt{\frac{d_\rho}{|G|}} \sum_{h \in G} \psi(h)\rho(h) = \sqrt{\frac{d_\rho |H|}{|G|}} \Pi_H^\rho$$

where $\Pi_H^\rho \triangleq |H|^{-1} \sum_{h \in H} \rho(h)$. It's easy to check that $(\Pi_H^\rho)^2 = \Pi_H^\rho$, i. e., Π_H^ρ is a projection operator. The probability of measuring a particular ρ using weak sampling is then

$$P_H(\rho) \triangleq \left\| \hat{\psi}(\rho) \right\|^2 = \frac{d_\rho |H|}{|G|} \mathbf{rk}\ \Pi_H^\rho \ . \tag{8}$$

This distribution takes a particularly nice form when H is a normal subgroup. Let $H^\perp$ denote the set of representations from $\hat{G}$ whose kernel contains H. The representations in $H^\perp$ (that is, representations which are trivial on H) are precisely the same as the representations of the quotient group G/H.

Lemma 1. *Let H be a normal subgroup of G. If ρ is an element of $H^\perp$, then the probability of observing ρ is equal to $d_\rho^2 |H|/|G|$; otherwise, the probability of observing ρ is zero.*

Proof. If $\rho \in H^\perp$, then for every $h \in H$, $\rho(h)$ is the identity operator. Since Π_H^ρ is simply the average of these, it is also equal to the identity operator, and thus has full rank. Hence

$$P_H(\rho) = \frac{d_\rho \mathbf{rk}\ \Pi_H^\rho |H|}{|G|} = \frac{d_\rho^2 |H|}{|G|} \ .$$

Now, if we add up the contributions to P_H from the representations in $H^\perp$, we have

$$\sum_{\rho \in H^\perp} P_H(\rho) = \sum_{\rho \in H^\perp} \frac{d_\rho^2 |H|}{|G|} = \frac{|H|}{|G|} \cdot \sum_{\rho \in \widehat{G/H}} d_\rho^2 = \frac{|H|}{|G|} \cdot |G/H| = 1 \ .$$

Hence the representations outside $H^\perp$ must contribute zero probability. □

We now show that one can reconstruct a normal subgroup H of an arbitrary finite group G in polynomial time, simply by sampling from the distribution P_H. Once enough samples have been produced, we then reconstruct the subgroup by intersecting the kernels of our sampled representations.

Theorem 1. *Let H be a normal subgroup of G. Let $\sigma_1, \ldots, \sigma_s$ be independent random variables sampled from the distribution P_H, with $s = c \log |G|$. Then*

$$Pr\left[H \neq \bigcap_i \ker \sigma_i \right] \leq e^{-\frac{(c-2)^2}{2c} \log |G|} \ .$$

Proof. Let $N_0 = G$, and let $N_i = \cap_{j=1}^{i} \ker \sigma_j$ be the intersection of the kernels sampled thus far. As they are intersections of normal subgroups, each N_i is normal in G. By Lemma 1, we cannot observe any representations except those whose kernel contains H, and thus $H \subset \ker \sigma_i$ for every i. Hence

$$H \subseteq N_s \subseteq N_{s-1} \subseteq \cdots \subseteq N_0 = G \ .$$

Our theorem rests on the fact that, with each new sample, we will make progress along the above chain with probability at least $1/2$; reaching H will thus take roughly $\log |G|$ many steps. We thus claim that if $N_i \neq H$, then $\Pr_{\sigma_{i+1} \in P_H}[N_{i+1} = N_i] \leq 1/2$. Indeed, by making use of Lemma 1 again, we see that

$$\begin{aligned} \Pr[N_{i+1} = N_i] = \Pr[N_i \subseteq \ker \sigma] &= \sum_{\rho \in N_i^{\perp}} d_\rho^2 \frac{|H|}{|G|} \\ &= |G/N_1| \cdot \frac{|H|}{|G|} = \frac{|H|}{|N_i|} \leq 1/2 \ . \end{aligned}$$

To complete the proof, we will need to apply a Chernoff bound. Let X_i be indicator random variables defined by $X_i = 1$ if $N_i = H$ or $N_i \neq N_{i-1}$ and zero otherwise. While the X_i are not necessarily independent, our claim above showed that $\Pr[X_i = 0 | \sigma_1, \ldots, \sigma_{i-1}] \leq 1/2$. We can thus define new independent random variables Y_i satisfying $\Pr[Y_i = 0] = 1/2$ and $\sum Y_i \leq \sum X_i$. By the Chernoff bound given in [11], $\Pr[\sum_i Y_i \leq (s-a)/2] < e^{-a^2/2s}$. This implies that

$$\Pr\left[\sum_i X_i \leq c \log |G|\right] < e^{-(c-2)^2 \log |G|/2c} \ .$$

Thus $\sum_i X_i \geq \log |G|$ with overwhelming probability; but in this case, $N_i \subsetneq N_{i-1}$ for every i, and hence $N_s = H$. □

We remark that the problem of reconstructing a normal subgroup by computing an intersection of representation kernels may, for general groups, be a very difficult problem. However, with the aid of a classical machine we can easily perform this task on abelian groups. By the structure theorem for finite abelian groups, we need only discuss the cyclic groups $G = \mathbb{Z}_n$. Now, suppose $\alpha \in \mathbb{Z}_n$ is a generator for the hidden subgroup in question, and that $\chi_{g_1}, \chi_{g_2}, \ldots, \chi_{g_s}$ are the sampled representations (that is, characters) of $\mathbb{Z}_n$. By Theorem 1 above, we know that

$$\bigcap_i \ker \chi_{g_i} = \langle \alpha \rangle \ .$$

We claim that the left hand side is simply $\ker \chi_g$ where $g = \gcd(g_1, g_2, \ldots, g_s)$. If $x \in \cap_i \ker \chi_{g_i}$, then

$$\chi_{g_i}(x) = e^{\frac{2\pi i g_i x}{n}} = 1 \ ,$$

and hence $g_i x \equiv 0 \bmod n$, for every i. By taking linear combinations, we have $gx \equiv 0 \bmod n$, and thus $x \in \ker \chi_g$. On the other hand, if x satisfies $gx \equiv 0 \bmod n$, then certainly $g_i x \equiv 0 \bmod n$ for every i, since the g_i are integer multiples of g. We have thus shown that $\langle \alpha \rangle = \ker \chi_g$. Along with Theorem 1, this proves that we can reconstruct hidden subgroups of cyclic groups efficiently by weak Fourier sampling a sufficient number of characters, and then computing their gcd using a classical machine.

5.2 Sieve Algorithms

5.2.1 Weak Sampling Fails

As we have shown, weak Fourier sampling (that is, measuring the representation name only) allows for the reconstruction of normal subgroups and, in particular, arbitrary subgroups of abelian groups. For some groups, however, this method cannot solve the HSP efficiently [11]. The following proposition, due to Alagic, Moore and Russell [1] shows that certain subgroups of *product groups* are indistinguishable using weak Fourier sampling alone. This is an example of a family of groups where *strong Fourier sampling* (that is, measuring the rows and columns in addition to the representation name) is necessary for resolving the HSP.

Proposition 1. *Let G be a group with an involution $\mu \notin Z(G)$, and let $H = \{1, m\} \leq G^n$ where m is chosen uniformly at random from the conjugacy class $[(\mu, \ldots, \mu)]$. Then the total variation distance between the weak Fourier sampling distributions (8) for the subgroups H and $\{1\}$ is at most $2^{-n/2}$.*

Proof. We upper bound the total variation distance between the distributions

in question:

$$\begin{aligned}
\left\|P_{\{1\}} - P_H\right\|_1 &= \sum_{\rho \in \widehat{G^n}} \left| \frac{d_\rho}{|G|^n} \mathbf{rk}\ \Pi^\rho_{\{1\}} - \frac{2d_\rho}{|G|^n} \mathbf{rk}\ \Pi^\rho_H \right| \\
&= \sum_{\rho \in \widehat{G^n}} \left| \frac{d_\rho}{|G|^n} \mathbf{rk}\ \Pi^\rho_{\{1\}} - \frac{2d_\rho}{|G|^n} \mathbf{tr} \left[\frac{\mathbb{1}_\rho + \rho(m)}{2} \right] \right| \\
&= \frac{1}{|G|^n} \sum_{\rho \in \widehat{G^n}} \left| d_\rho^2 - d_\rho^2 \left(1 + \frac{\chi_\rho(m)}{d_\rho} \right) \right| \\
&= \frac{1}{|G|^n} \sum_{\rho \in \widehat{G^n}} \left| d_\rho \cdot \chi_\rho(m) \right| \ .
\end{aligned}$$

Viewing the last line as an inner product, we apply Cauchy-Schwarz to get

$$\begin{aligned}
\left\|P_{\{1\}} - P_H\right\|_1 &\leq \frac{1}{|G|^n} \left(\sum_\rho d_\rho^2 \right)^{1/2} \left(\sum_\rho \chi_\rho(m) \chi_\rho^*(m) \right)^{1/2} \\
&= \frac{1}{|G|^{n/2}} \left(\sum_{\rho \in \widehat{G^n}} \chi_\rho(m) \chi_\rho^*(m) \right)^{1/2} \\
&= \frac{1}{|G|^{n/2}} \left(\sum_{\rho \in \widehat{G}} \chi_\rho(\mu) \chi_\rho^*(\mu) \right)^{n/2} \ .
\end{aligned}$$

Here we have used the fact that the character of an irreducible G^n-representation is an n-fold product of characters of irreducible G-representations. The term $\sum_\rho \chi_\rho(\mu)\chi_\rho^*(\mu)$ is in fact the character of the so-called *conjugation representation* of G; this is the representation defined on $\mathbb{C}G$ by linearly extending the rule $g \cdot x \mapsto gxg^{-1}$. The character of this representation, evaluated at μ, is exactly the number of fixed points of the conjugation action of μ on G, i. e., the size of the centralizer C_μ. As μ is not in the center, C_μ is a proper subgroup, and hence $\chi_C(\mu) \leq |G|/2$, which completes the proof.

□

5.2.2 Sieve Algorithm Sketch

Recent negative results have shown that even strong Fourier sampling is insufficient to efficiently resolve the HSP on certain highly nonabelian groups. For instance, Hallgren, Moore, Rötteler, Russell and Sen [10] showed that

multiregister Fourier sampling over $\Omega(\log |G|)$ registers is required to efficiently distinguish subgroups of certain families of groups; these families include the symmetric groups (presumably critical for the Graph Isomorphism application discussed in Section 2.3), and the nonabelian direct product groups. Despite this, we can still do provably better than classically for certain groups to which these negative results apply. Indeed, using a *representation sieve* idea pioneered by Kuperberg [14] for the HSP on dihedral groups, a subexponential-time sieve algorithm of Alagic, Moore and Russell [2] resolves the HSP on direct product groups. These are groups of the form $G = K^n$ where K is nonabelian and of constant size. We now discuss, in brief, the central idea behind algorithms based on Kuperberg's representation sieve.

Recall that the result of weak Fourier sampling is essentially a "state vector" lying in an irreducible representation of our group G. Now, if we perform weak sampling twice, we will have two state vectors lying in irreducibles which we will label ρ and σ. The representations ρ and σ naturally define another representation $\rho \otimes \sigma$ on the tensor product $V_\rho \otimes V_\sigma$ of their respective spaces. This is done by the "diagonal action":

$$[\rho \otimes \sigma](g) \triangleq \rho(g) \otimes \sigma(g) \ .$$

In general, this new representation is not irreducible; it does, however, have an irreducible decomposition

$$\rho \otimes \sigma \cong \bigoplus_j \tau_j \ .$$

The essential ingredient of the aforementioned sieve algorithms is this: given state vectors in ρ and σ, we can use a modified form of weak sampling to produce a state vector in one of the constituents τ_j of $\rho \otimes \sigma$. Moreover, our knowledge of the representation theory of the underlying group gives us a nice picture of which τ_j our new state vector may lie in; for instance, if the hidden subgroup is actually trivial, then the resultant state will lie in τ_j with probability $d_{\tau_j}/d_{\rho\otimes\sigma} = d_{\tau_j}/(d_\rho d_\sigma)$. A sieve algorithm may thus proceed as follows:

1. use weak Fourier sampling to generate a large (but still subexponential) collection of states, each lying in some irreducible representation;

2. cleverly pair up the states generated above, and sample new states from the tensor products of these pairs, now lying in representations of *smaller dimension* (on average);

3. repeat (2) until you generate states lying in one-dimensional representations.

Once we have generated sufficiently many states in one-dimensional representations, we can then reconstruct the hidden subgroup; this is roughly analogous to the abelian case, where all irreducibles are one-dimensional. We remark that the repeated applications of step (2) constitute a (rather complicated) multiregister measurement over all of the registers containing the coset states initially sampled in step (1).

5.3 The pretty-good measurement

Recall from the discussion preceding Equation (5) that typical approaches to the hidden subgroup problem involve production of *coset states* of the form

$$\frac{1}{\sqrt{|H|}} \sum_{h \in H} |ch\rangle \ ,$$

where c is an independently chosen element in G. This "state" is really a classical probability distribution over the sorts of quantum states we have described in the article thus far. Such objects arise anytime a measurement takes place, and are called *mixed states*. (The quantum states we have discussed previously, unit length vectors, are the special case with a trivial probability distribution and are called *pure states* when it is useful to emphasize the distinction.)

It turns out that there are group/subgroup combinations where this state contains very little information about H [18]. On the other hand, it is known that polynomially many (in $\log |G|$) of these states do, at least in principle, completely determine the subgroup H [9]. The problem, from a quantum computational perspective, is to discover an efficient means of extracting this information from the mixed state.

The problem of "identifying" a given state from a known list of possibilities has a long history in quantum mechanics, though most previous work was less concerned with computational efficiency than it was with the information-theoretic aspects of the problem: that is, whether or not measurements even existed to tease apart various families of states.

A remarkable discovery of Bacon, Childs, and van Dam [3] is that a *generic* measurement, the "pretty-good measurement," that one can define for any fixed family of mixed states can actually be efficiently implemented in some cases of interest for the hidden subgroup problem. Furthermore, they show that these measurements are powerful enough to distinguish the subgroups, giving rise to efficient solutions to the hidden subgroup problem

for specially structured groups. (It was later shown that with sufficiently many copies of the mixed state above, the pretty good measurement is always a rich enough measurement to distinguish hidden subgroups [12].)

6 Conclusion

The hidden subgroup problem (and the symmetry finding problem, in general) unite quantum computation, a handful of classical computational problems, and aspects of the representation theory of finite groups. We indicated, in Section 5, the principal algorithmic techniques that have been developed for attacking this problem in general: Fourier sampling, sieve methods, and implementation of the pretty good measurement. In all of these cases, the essential insight on which the algorithm depends illuminates the connection between coset states and the algebraic structure of the group's representation theory. Despite this progress, the guiding problem in the area of nonabelian hidden subgroup problems—Graph Isomorphism—appears to be quite immune to known techniques [17, 19]. Happily, the area has seen rapid development in the last 5 years on the algorithmic, information-theoretic, and representation-theoretic fronts; we can be sure that the area and its hallmark problem have more secrets to uncover.

References

[1] Gorjan Alagic, Cristopher Moore, and Alexander Russell. Strong Fourier sampling fails over G^n. Technical Report quant-ph/0511054, arXiv.org e-Print archive, 2005.

[2] Gorjan Alagic, Cristopher Moore, and Alexander Russell. Quantum algorithms for Simon's problem over general groups. In Nikhil Bansal, Kirk Pruhs, and Clifford Stein, editors, *Proc. of SODA 2007*, pages 1217–1224. SIAM, 2007.

[3] Dave Bacon, Andrew M. Childs, and Wim van Dam. From optimal measurement to efficient quantum algorithms for the hidden subgroup problem over semidirect product groups. In *Proc. of FOCS 2005* [13], pages 469–478.

[4] Robert Beals. Quantum computation of fourier transforms over symmetric groups. In *Proc. of STOC '97*, pages 48–53, New York, NY, USA, 1997. ACM Press.

[5] Ethan Bernstein and Umesh V. Vazirani. Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411–1473, 1997.

[6] Andrew Childs, Leonard Schulman, and Umesh Vazirani. Quantum algorithms for hidden nonlinear structures. Technical Report quant-ph/0705.2784, arXiv.org e-Print archive, 2007.

[7] Thomas Decker, Jan Draisma, and Pawel Wocjan. Quantum algorithm for identifying hidden polynomial function graphs. Technical Report quant-ph/0706.1219, arXiv.org e-Print archive, 2007.

[8] David Dummit and Richard Foote. *Abstract Algebra*. John Wiley & Sons, Hoboken, NJ, third edition, 2004.

[9] Mark Ettinger, Peter Høyer, and Emanuel Knill. The quantum query complexity of the hidden subgroup problem is polynomial. *Inf. Process. Lett.*, 91(1):43–48, 2004.

[10] Sean Hallgren, Cristopher Moore, Martin Rötteler, Alexander Russell, and Pranab Sen. Limitations of quantum coset states for graph isomorphism. In Jon M. Kleinberg, editor, *Proc. of STOC 2006*, pages 604–617. ACM, 2006.

[11] Sean Hallgren, Alexander Russell, and Amnon Ta-Shma. Normal subgroup reconstruction and quantum computation using group representations. In *Proc. of STOC 2000*, pages 627–635. ACM, 2000.

[12] Masahito Hayashi, Akinori Kawachi, and Hirotada Kobayashi. Quantum measurements for hidden subgroup problems with optimal sample complexity. Technical Report quant-ph/0604174, arXiv.org e-Print archive, 2006.

[13] IEEE. *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*. IEEE, 2005.

[14] Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM J. Comput.*, 35(1):170–188, 2005.

[15] Cristopher Moore, Daniel Rockmore, and Alexander Russell. Generic quantum fourier transforms. In *Proc. of SODA '04*, pages 778–787, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics.

[16] Cristopher Moore, Daniel N. Rockmore, Alexander Russell, and Leonard J. Schulman. The power of basis selection in fourier sampling: Hidden subgroup problems in affine groups. In J. Ian Munro, editor, *Proc. of SODA 2004*, pages 1113–1122. SIAM, 2004.

[17] Cristopher Moore, Alexander Russell, and Leonard Schulman. Tight results on multiregister fourier sampling: Quantum measurements for graph isomorphism require entanglement. Technical Report quant-ph/0511149, arXiv.org e-Print archive, 2006.

[18] Cristopher Moore, Alexander Russell, and Leonard J. Schulman. The symmetric group defies strong Fourier sampling. In *Proc. of FOCS 2005* [13], pages 479–490.

[19] Cristopher Moore, Alexander Russell, and Piotr Sniady. On the impossibility of a quantum sieve algorithm for graph isomorphism. In *STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 536–545, New York, NY, USA, 2007. ACM Press.

[20] Peter W. Shor. Polynominal time algorithms for discrete logarithms and factoring on a quantum computer. In Leonard M. Adleman and Ming-Deh A. Huang, editors, *Proc. of ANTS 1994*, volume 877 of *Lecture Notes in Computer Science*, page 289. Springer, 1994.

[21] Daniel R. Simon. On the power of quantum computation. *SIAM J. Comput.*, 26(5):1474–1483, 1997.

[22] Wim van Dam, Sean Hallgren, and Lawrence Ip. Quantum algorithms for some hidden shift problems. In *Proc. of SODA 2003*, pages 489–498. ACM, 2003.