SIAM J. COMPUT.
Vol. 43, No. 1, pp. 1–24

HIDDEN TRANSLATION AND TRANSLATING COSET IN QUANTUM COMPUTING[*]

KATALIN FRIEDL[†], GÁBOR IVANYOS[‡], FRÉDÉRIC MAGNIEZ[§], MIKLOS SANTHA[¶], AND PRANAB SEN[‖]

Abstract. We give efficient quantum algorithms for the problems of HIDDEN TRANSLATION and HIDDEN SUBGROUP in a large class of nonabelian solvable groups, including solvable groups of constant exponent and of constant length derived series. Our algorithms are recursive. For the base case, we solve efficiently HIDDEN TRANSLATION in $\mathbb{Z}_p^n$, whenever p is a fixed prime. For the induction step, we introduce the problem TRANSLATING COSET generalizing both HIDDEN TRANSLATION and HIDDEN SUBGROUP and prove a powerful self-reducibility result: TRANSLATING COSET in a finite solvable group G is reducible to instances of TRANSLATING COSET in G/N and N, for appropriate normal subgroups N of G. Our self-reducibility framework, combined with Kuperberg's subexponential quantum algorithm for solving HIDDEN TRANSLATION in any abelian group, leads to subexponential quantum algorithms for HIDDEN TRANSLATION and HIDDEN SUBGROUP in any solvable group.

Key words. quantum algorithms, hidden subgroup problem, solvable groups

AMS subject classification. 68Q25

DOI. 10.1137/130907203

1. Introduction.

Quantum computing is an extremely active research area (for introductions, see, e.g., [30, 1, 36, 35]). Many of the superpolynomial speedups achieved by quantum algorithms over their best known classical counterparts have been in a group theoretical setting. In this setting, we are given a finite group G and, besides the group operations, we also have at our disposal a function f mapping G into a finite set. The function f can be queried via an oracle. The time complexity of an algorithm is measured by the overall running time, including both the queries (counting a query as one step) and the quantum and/or classical processing of these queries. The most important unifying problem of group theory for the purpose of quantum algorithms has turned out to be HIDDEN SUBGROUP, which can be cast in the following broad terms: Let H be a subgroup of G such that f is constant on each left coset of H and distinct on different left cosets. We say that f *hides* the subgroup H. The task is to determine the *hidden subgroup* H.

[*] Received by the editors January 24, 2013; accepted for publication (in revised form) September 13, 2013; published electronically January 2, 2014. A preliminary version of this paper appeared in [15]. This work was partially supported by the European Commission IST STREP projects Quantum Computer Science (QCS) 255961 and Quantum Algorithms (QALGO) 600700, by the French ANR Blanc program under contract ANR-12-BS02-005 (RDAM project), and by the Hungarian National Science Fund (OTKA) through grants NK105645 and K77476. Research at the Centre for Quantum Technologies is funded by the Singapore Ministry of Education and the National Research Foundation, and also through the Tier 3 Grant "Random Numbers from Quantum Processes."

http://www.siam.org/journals/sicomp/43-1/90720.html

[†] Budapest University of Technology and Economics, Budapest, Hungary (friedl@cs.bme.hu).

[‡] Institute for Computer Science and Control, Hungarian Academy of Sciences, Budapest, Hungary (gabor.ivanyos@sztaki.hu).

[§] CNRS, LIAFA, Université Paris Diderot, Sorbonne Paris-Cité, Paris, France 75205 (frederic.magniez@univ-paris-diderot.fr).

[¶] CNRS, LIAFA, Université Paris Diderot, Sorbonne Paris-Cité, Paris, France 75205, and Centre for Quantum Technologies, National University of Singapore, Singapore (miklos.santha@liafa.univ-paris-diderot.fr).

[‖] School of Technology and Computer Science, Tata Institute of Fundamental Research, Mumbai, India 400005 (pgdsen@tcs.tifr.res.in).

While no classical algorithm can solve this problem with polynomial query complexity even if G is abelian, the biggest success of quantum computing until now is that it can be solved by a quantum algorithm efficiently for any abelian G. We will refer to this quantum algorithm as the standard algorithm for HIDDEN SUBGROUP. The main tool for this solution is Fourier sampling based on the (approximate) quantum Fourier transform for abelian groups which can be efficiently implemented quantumly [29]. Simon's XOR-mask finding [42], Shor's factorization and discrete logarithm finding algorithms [41], and Kitaev's algorithm [29] for the abelian stabilizer problem are all special cases of this general solution. Quantum algorithms of Hallgren [20, 21] and Schmidt and Vollmer [40] computing class groups and unit groups of number fields, including the solution of Pell's equation, also follow along these lines.

Finding an efficient algorithm for HIDDEN SUBGROUP for nonabelian groups G is considered to be one of the most important challenges at present in quantum computing. Besides its intrinsic mathematical interest, the importance of this problem is enhanced by the fact that it contains as a special case the graph isomorphism problem. Unfortunately, although its query complexity is shown to be polynomial by Ettinger, Høyer, and Knill [14], nonabelian HIDDEN SUBGROUP seems to be much more difficult than the abelian case. Although considerable effort was spent on it in the last few years, only a small number of successes can be reported. They can be divided into two categories. The standard abelian Fourier sampling based algorithm has been extended to some nonabelian groups in [39, 22, 19, 16, 33, 12] using the quantum Fourier transform over these (nonabelian) groups. Although efficient quantum Fourier transform implementations are known for several nonabelian groups [8, 23, 37, 32], the power of the technique appears to be very limited. In a different approach, HIDDEN SUBGROUP was efficiently solved in the context of specific nonabelian black-box groups [5, 45] by [26] without using the Fourier transform on the group, and instead using Fourier transforms over abelian groups only. Similarly, only abelian Fourier transforms were used by [24, 6, 10, 27, 28] to solve the hidden subgroup problem in some specific kinds of nonabelian groups. See [11] for a more detailed review of hidden subgroup algorithms and related problems.

In light of the apparent hardness of HIDDEN SUBGROUP in nonabelian groups, a natural line of research is to address subproblems of HIDDEN SUBGROUP which, in some groups, capture the main difficulty of the original problem. In a pioneering paper, Ettinger and Høyer [13], in the case of dihedral groups, implicitly considered another paradigmatic group problem, HIDDEN TRANSLATION. Here we are given two injective functions f_0 and f_1 from a finite group G to some finite set such that, for some group element u, the equality $f_1(xu) = f_0(x)$ holds for every x. The task is to find the *translation* u. In fact, whenever G is abelian, HIDDEN TRANSLATION is an instance of HIDDEN SUBGROUP in the semidirect product $G \rtimes \mathbb{Z}_2$, where the hiding function is $f(x, b) = f_b(x)$. The group action in $G \rtimes \mathbb{Z}_2$ is defined as $(x_1, b_1)\cdot(x_2, b_2) = (x_1 + (-1)^{b_1} x_2, b_1 \oplus b_2)$, where $+$ denotes the group operation in G and $\oplus$ denotes the group operation in $\mathbb{Z}_2$. In $G \rtimes \mathbb{Z}_2$, f hides the subgroup $H = \{(0, 0), (u, 1)\}$. Actually, there is an efficient quantum reduction in the other direction as well, and the two problems are quantum polynomial time equivalent [13]. A nice consequence of this equivalence is that instead of dealing with HIDDEN SUBGROUP in the nonabelian group $G \rtimes \mathbb{Z}_2$, we can address HIDDEN TRANSLATION in the abelian group G. Ettinger and Høyer [13] have shown that HIDDEN TRANSLATION can be solved by a two-step procedure when $G = \mathbb{Z}_N$ is cyclic: a polynomial number of Fourier samplings over the abelian group $\mathbb{Z}_N \times \mathbb{Z}_2$ followed by an exponential time classical stage without further queries. The best known quantum algorithm for HIDDEN TRANSLATION in cyclic (and, in general, abelian) groups is Kuperberg's subexponential time method [31]. Its relation to certain lattice problems investigated by Regev [38] provides evidence that Hidden Translation in cyclic groups might in fact be difficult.

In a related work, van Dam, Hallgren, and Ip [44] gave efficient solutions for three cases of what they call the hidden shift problem. They also define another problem called the hidden coset problem which generalizes hidden shift. Their hidden coset problem can be viewed as a generalization of our Hidden Translation to not necessarily injective functions. While their paper gives efficient quantum algorithms for some specific hidden coset problems, in general the hidden coset problem is of exponential query complexity even in $\mathbb{Z}_2^n$.

Our first result (Theorem 3.5) is an efficient quantum algorithm for Hidden Translation in the case of elementary abelian p-groups, that is, groups $\mathbb{Z}_p^n$, for any fixed prime number p. The quantum part of our algorithm is the same as in Ettinger and Høyer's procedure [13]: it consists of performing Fourier sampling over the abelian group $\mathbb{Z}_p^n \times \mathbb{Z}_2$. But while their classical postprocessing requires exponential time, here we are able to recover classically the translation in polynomial time from the samples. It turns out that Fourier sampling produces vectors y nonorthogonal to the translation u; that is, we obtain linear inequations for the unknown u. This is different from the situation in the standard algorithm for the abelian Hidden Subgroup, where only vectors orthogonal to the hidden subgroup are generated. We show that, after a polynomial number of samplings, the system of linear inequations has a unique solution with high probability, which we are able to determine in deterministic polynomial time. An immediate consequence of Theorem 3.5 is that Hidden Subgroup in $\mathbb{Z}_p^n \rtimes \mathbb{Z}_2$ is efficiently solvable by a quantum algorithm.

To solve Hidden Translation in other groups (which include abelian groups of constant exponent), we embark in a radically new direction whose basic idea is *self-reducibility*. Since Hidden Translation is not well-suited for this self-reducibility based approach, we define a new paradigmatic group problem. Notice that there is a natural combination of Hidden Translation with Hidden Subgroup. This is the version of Hidden Translation where the functions f_0 and f_1 are not necessarily injective, but they are certain subgroup hiding functions. Indeed, if f_1 hides a subgroup H and $f_0(x) = f_1(xu)$ for some $u \in G$ and for every $x \in G$, then the set of all such elements u form a right coset of H. (In the context of graph isomorphisms, the corresponding problem would be determining all the bijections between the vertex sets which are isomorphisms. This set is a coset of the automorphism group of one of the graphs.) The self-reducibility will be based on "averaging" over normal subgroups so that we actually get a problem over the factor group. We will give an averaging procedure which results in quantum superpositions. Therefore our new problem, called Translating Coset, is a combination of Hidden Translation and Hidden Subgroup where we have quantum states as input.[1] Translating Coset also involves quantum group actions, that is, groups acting on a finite set of mutually orthogonal quantum states. Given two such states, $|\phi_0\rangle$ and $|\phi_1\rangle$, the Translating Coset problem consists of finding their *translating coset*, which is defined to be the stabilizer subgroup of $|\phi_1\rangle$ and a group element that maps $|\phi_1\rangle$ to $|\phi_0\rangle$.

It turns out that with a slight modification, our algorithm of Theorem 3.5 also works for Translating Coset in $\mathbb{Z}_p^n$ whenever many copies of the input states

[1] In the preliminary version [15] of the present paper, the problem Translating Coset was called Orbit Coset. This was due to the fact that the problem is actually a constructive version of testing membership in orbits of permutation groups.

are given. Moreover, we show that TRANSLATING COSET has the following self-reducibility property in any finite solvable group G: it is reducible to instances of TRANSLATING COSET in G/N and N for any normal subgroup $N \lhd G$ (Theorem 4.11). This is the first general self-reducibility result for a problem subsuming HIDDEN SUBGROUP. The proof of the result involves a new technique which is based upon constructing the uniform superposition of the orbit of a given quantum state (ORBIT SUPERPOSITION). The importance of generating specific superpositions for solving important algorithmic problems has been observed before; see, for instance, the paper of Aharonov and Ta-Shma [3]. For example, generating the uniform superposition of all graphs isomorphic to a given graph, which in fact is an instance of the ORBIT SUPERPOSITION problem of the symmetric group S_n acting on an n-vertex graph, would allow us to solve the graph isomorphism problem. We show how ORBIT SUPERPOSITION is related to TRANSLATING COSET (Theorem 4.10). The self-reducibility of TRANSLATING COSET combined with its solvability for $\mathbb{Z}_p^n$ enables us to design an efficient quantum algorithm for TRANSLATING COSET in groups that we call smoothly solvable groups (Theorem 4.16). These groups include solvable groups of constant exponent and constant length derived series, in particular, unitriangular matrix groups of constant dimension over finite fields of constant characteristic. For the special case of STABILIZER (i.e., TRANSLATING COSET when $|\phi_1\rangle = |\phi_0\rangle$), we obtain an efficient quantum algorithm for an even larger class of solvable groups, i.e., for solvable groups having a smoothly solvable commutator subgroup (Theorem 4.16). As an immediate consequence, we get efficient quantum algorithms for HIDDEN TRANSLATION and HIDDEN SUBGROUP in the same groups as TRANSLATING COSET and STABILIZER, respectively. By combining our self-reducibility results above with Kuperberg's subexponential time algorithm for HIDDEN TRANSLATION in abelian groups [31], and using the fact that every solvable group G has derived series of length $\mathrm{O}(\log\log|G|)$ [17], we get subexponential time algorithms for HIDDEN TRANSLATION and HIDDEN SUBGROUP in all solvable groups (Theorem 4.18), and quasi-polynomial time quantum algorithm for HIDDEN TRANSLATION and HIDDEN SUBGROUP in solvable groups of constant exponent (Theorem 4.17).

2. Preliminaries.

2.1. Quantum computation background.

For a background on standard quantum computing, we refer the reader to [35, 30]. We will consider problems whose inputs and outputs might be either classical or quantum. Moreover most of our problems are promise problems where a part of the input is given by an oracle. A *problem* is a relation $\mathcal{P} \subseteq I \times O$, where I is the set of *inputs*, and O the set of possible *outputs*. For a family of functions $\mathcal{F}$, an *oracle problem* is a family of relations $(\mathcal{P}^f)_{f\in\mathcal{F}}$, where f ranges over the family $\mathcal{F}$. The function f is given by a quantum oracle, that is, a unitary matrix U_f implementing the map $U_f|x\rangle|0\rangle = |x\rangle|f(x)\rangle$.

For any finite set S, we denote by $|S\rangle$ the uniform superposition of elements in S: $|S\rangle = \frac{1}{\sqrt{|S|}}\sum_{x\in S}|x\rangle$ when $S \neq \emptyset$, and $|S\rangle = |\emptyset\rangle$ when $S = \emptyset$, where $|\emptyset\rangle$ is a specific basis element.

A quantum algorithm is a quantum circuit consisting of a succession of quantum gates. Sometimes we describe quantum algorithms using intermediate measurements, but they can always be replaced by unitary operations acting on the system plus ancilla qubits [2]. The *output state* of the algorithm is defined to be the reduced state at the end of the algorithm of a special register of qubits, called the *output register*.

Namely, the output state of the algorithm is obtained by tracing out all but the qubits of the output register at the end of the algorithm.

In this paper, we consider problems with many possible correct answers. For example, an algorithm for HIDDEN SUBGROUP is said to be correct if it outputs any generating set for the hidden subgroup. Therefore we say that a quantum algorithm or a unitary transformation *solves* a problem P *with error* ε if for every input $i \in I$ it produces an output state whose trace distance is at most ε from some mixture over $\{o \in O : (i, o) \in \mathcal{P}\}$ (see, e.g., [2] for a definition of trace distance).

The *time complexity* of an algorithm is the number of gates and oracle calls in the circuit. For every problem, the *input size* is the number of classical or quantum bits of an input. We say that a computational problem can be solved in *quantum time* $t(n)$ if there exists a quantum algorithm which solves the problem with bounded error in time $t(n)$, where n is the input size.

2.2. Group theory background. Recall that the *exponent* of a finite group is the least common multiple of the order of its elements, and an *elementary abelian group* is a group isomorphic to $\mathbb{Z}_p^n$ for some positive integer n and for some prime p. Obviously, the exponent of $\mathbb{Z}_p^n$ is p. Let G be a finite group. If X is a subset of G, then $\langle X\rangle$ denotes the subgroup of G generated by X.

2.2.1. Black-box groups. Our results concern groups represented in the general framework of black-box groups [5, 45] with unique encoding. In this model, the elements of a finite group G are uniquely encoded by binary strings of length ℓ, and the group operations are performed by an oracle (the black box). The group is given in terms of a collection of generators, and the oracle may actually define operations for a potentially larger group. We formally denote the encoding by a mapping enc from G to $\{0,1\}^\ell$. For quantum algorithms, the group operations are performed using a reversible oracle; see [45] for a detailed description. The *encoding length* ℓ has to be at least $\log|G|$, and is usually $\mathrm{O}(\log|G|)$. We measure the running time of our algorithm in terms of the input size ℓ. Several times in this paper we will be dealing with subgroups or factor groups of black-box groups wherein we will still continue to measure the running time in terms of the input length ℓ for the original group G, since we continue to use the original encodings for the subgroup elements. But even in this case, all the encoding lengths for all subgroups shall be $\mathrm{O}(\log|G|)$, where G is the original group.

We do assume in all our problems that the groups are input by at most $\log|G|$ generators. This is legitimate as there are several efficient methods, e.g., the quantum algorithms given in [46] or [26] that produce at most $\log|G|$ generators for a solvable black-box group G, even if it is given by a larger set of generators. The input size corresponding to G is set to ℓ, instead of $\ell \times \log|G|$, for convenience.

2.2.2. Solvable groups. A sequence $G_0 \geq G_1 \geq \cdots \geq G_m$ of subgroups is a *subnormal series* of G if each G_i is a normal subgroup of G_{i-1}. We use the notation $G_0 \rhd G_1 \rhd \cdots \rhd G_m$ for a subnormal series. The *length* of such a series is m.

The group G is a *solvable* group when there exists a subnormal series $G_0 \rhd G_1 \rhd \cdots \rhd G_m$ such that $G = G_0$, $G_m = \{1_G\}$ and the factors G_i/G_{i+1} $(i = 0, 1, \ldots, m-1)$ are abelian.

A natural way of constructing a subnormal series of the solvable group G is to consider its derived subgroups. For any group H, let us first define and denote the *commutator subgroup* H' of H by $H' = \langle\{h^{-1}k^{-1}hk : h, k \in H\}\rangle$. Then the *derived subgroups* $G^{(i)}$ $(i = 0, 1, 2, \ldots)$ are defined by induction: $G^{(0)} = G$; and the

$(i+1)$th derived subgroup $G^{(i+1)}$ is defined as the commutator $(G^{(i)})'$ of $G^{(i)}$. All the subgroups $G^{(i)}$ are normal subgroups of $G^{(j)}$ for $0 \leq j < i$. Clearly the group G is solvable if $G^{(d)} = \{1_G\}$ for some positive integer d and the *derived length* of G is the smallest such integer d. The *derived series* of a solvable group G is the chain $G = G^{(0)} \triangleright G^{(1)} \triangleright \cdots \triangleright G^{(d)} = \{1_G\}$.

In the case of an abelian group G, we have at our disposal [9] an efficiently computable isomorphism for the *cyclic decomposition* $\theta : \mathbb{Z}_{p_1^{k_1}} \times \cdots \times \mathbb{Z}_{p_r^{k_r}} \to G$, where $p_i^{k_i}$ are prime powers for primes p_i. Whenever G is solvable, the decomposition of G into its derived series can be computed by a classical randomized procedure [4].

2.2.3. Smooth groups. We introduce a shorthand terminology for the specific class of solvable groups for which our method works in polynomial time. We say that an abelian group G is (e,s)-*smooth* if it has a subgroup N of index at most s with exponent at most e. A subnormal series $G = G_0 \triangleright G_1 \triangleright \cdots \triangleright G_m = \{1_G\}$ of a solvable group G is (e,s)-*smooth* if each factor group G_{i-1}/G_i is (e,s)-smooth. A solvable group G is (e,s)-*smooth* if its derived series is (e,s)-smooth.

The methods of this paper will work in polynomial time for (e,s)-smooth solvable groups G with constant derived length and with constant e and $s = \text{poly}(\log|G|)$. We introduce the shorthand terminology *smoothly solvable* for such groups. Solvable groups having constant derived length and satisfying the property that the factors of the consecutive derived subgroups are of exponent bounded by a constant are the most typical examples of smoothly solvable groups. An example of such a solvable group is a unitriangular matrix group of constant dimension over a finite field of constant characteristic.

2.2.4. Quantum Fourier sampling. When G is a finite abelian group, we identify with G the set $\widehat{G}$ of characters of G via some fixed isomorphism $y \mapsto \chi_y$. (For a group G isomorphic to $\mathbb{Z}_k^n$, it is usual to define $\chi_y(x)$ as $e^{\frac{2\pi i}{k} x \cdot y}$, where $x \cdot y$ stands for the standard inner product $\sum_{i=1}^n x_i y_i \pmod{k}$. Of course, this definition requires—and depends on—an isomorphism of G with $\mathbb{Z}_k^n$.) The *orthogonal subgroup of* $H \leq G$ is defined as $H^\perp = \{y \in G : \forall h \in H, \chi_y(h) = 1\}$. The *quantum Fourier transform* over G is the unitary transformation defined for every $x \in G$ by $\text{QFT}_G|x\rangle = \frac{1}{\sqrt{|G|}} \sum_{y \in G} \chi_y(x)|y\rangle$. For the sake of convenience, we will use the exact abelian quantum Fourier transform in our algorithm. Actual implementations [29, 34] introduce only exponentially small errors.

The following well-known quantum Fourier sampling algorithm will be used as a building block, where G is a finite abelian group, S is a finite set, and $f : G \to S$ is given by a quantum oracle. This algorithm is actually the main ingredient for solving Hidden Subgroup in abelian groups when the function f hides a subgroup $H \leq G$. In that case, `FourierSampling`$^f(G)$ generates the uniform distribution over $H^\perp$. In the algorithm, $|0\rangle_S$ stands for an arbitrary but fixed element of S.

```
FourierSampling$^f(G)$
  1. Create state $\frac{1}{\sqrt{|G|}} \sum_{x \in G} |x\rangle|0\rangle_S$.
  2. Query function $f$.
  3. Compute $\text{QFT}_G$ on first register.
  4. Measure and output the first register.
```

A function $f : G \to \mathbb{C}^S$ is a *quantum function* if, for every $x \in G$, the vector $|f(x)\rangle$ has unit norm and, for every $x, y \in G$, the vectors $|f(x)\rangle$ and $|f(y)\rangle$ are either the same or orthogonal. We say that the quantum function f is *given* by a quantum oracle if we have at our disposal a unitary transformation U_f and its inverse U_f^{-1} satisfying $U_f|x\rangle|0\rangle = |x\rangle|f(x)\rangle$ for every $x \in G$.

2.2.5. Order finding and generalized discrete logarithm. We also assume for simplicity that we have at our disposal a zero-error quantum algorithm for computing the generalized discrete logarithm and for order finding. Given a basis $h_1, h_2, \ldots, h_l$ of an abelian group H and $h \in H$, the *generalized discrete logarithm* consists of finding nonnegative integers $\alpha_1, \alpha_2, \ldots, \alpha_l$ such that $h = h_1^{\alpha_1} h_2^{\alpha_2} \ldots h_l^{\alpha_l}$. Given a group element g in any group, *order finding* consists of finding the smallest positive integer r such that g^r is the identity element.

The actual implementations for period finding [41], for the single basis element case of discrete logarithm [41] and for the general case [26], introduce only exponentially small errors. Note that for discrete logarithm, one can also use a generalization of the single basis element case by [34] which runs without error if one has access to single qubit rotation gates of arbitrary precision.

2.3. The problems. Here we define the problems we are dealing with. Each problem is parametrized by some fixed group, and potentially by some group action. These are given, as we specified above, by oracles. Some inputs, usually functions on the group, can also be given by oracles; we will refer to them as *oracle inputs*.

Let G be a finite group and let f_0, f_1 be two injective functions from G to some finite set S. The pair of functions (f_0, f_1) can equivalently be considered as a single function $f : G \times \mathbb{Z}_2 \to S$, where by definition $f(x, b) = f_b(x)$. We will use f for (f_0, f_1) when it is convenient in the coming discussion. We call an element $u \in G$ the *translation* of f if for every $x \in G$ we have $f_1(xu) = f_0(x)$.

Hidden Translation(G)
Oracle input: Two injective functions f_0, f_1 from G to some finite set S such that $f = (f_0, f_1)$ has a translation $u \in G$.
Output: u.

For a finite group G and a finite set Γ of mutually orthogonal quantum states, we consider group actions of G on Γ. By definition, $\alpha : G \times \Gamma \to \Gamma$ is a *group action* if for every $x \in G$ the quantum function $\alpha_x : |\phi\rangle \mapsto |\alpha(x, |\phi\rangle)\rangle$ is a permutation over Γ, such that the map $x \mapsto \alpha_x$ is a homomorphism from G to the symmetric group on Γ, i.e., α_{1_G} is the identity map and $\alpha_x \circ \alpha_{y^{-1}} = \alpha_{xy^{-1}}$ for every $x, y \in G$. We extend α linearly to superpositions over Γ. (The condition that G permutes the orthonormal system Γ of states is essential; we do not consider general unitary actions G on Hilbert spaces.) When the group action α is fixed, we use the notation $|x \cdot \phi\rangle$ for the state $|\alpha(x, |\phi\rangle)\rangle$. Having a group action α at our disposal means having a quantum oracle realizing the unitary transformation $|x\rangle|\phi\rangle \mapsto |x\rangle|x \cdot \phi\rangle$. For any positive integer t, we denote by α^t the group action of G on $\Gamma^t = \{|\phi\rangle^{\otimes t} : |\phi\rangle \in \Gamma\}$ defined by $\alpha^t(x, |\phi\rangle^{\otimes t}) = |x \cdot \phi\rangle^{\otimes t}$. Observe that one can construct a quantum oracle for α^t using t queries to a quantum oracle for α. We need the notion of α^t for the following reason. Below, we define problems involving group actions on quantum superpositions where the input superpositions cannot, in general, be cloned (that is, it may be impossible to make further copies of the input state from just one). However, it will be possible to generate multiple independent copies of the input superpositions by a separate process before the start of our algorithm. Hence, in the interests of reducing the error of our algorithm, we start it off with several independent copies of the input superpositions. Our self-reducibility arguments will reduce the main problem into a bunch of problems involving actions of smaller groups on quantum superpositions. To solve each of these subproblems with small error, we will require that the self-reduction process leave a sufficient number of independent copies of the input superpositions for a subproblem. This is easy to ensure since we start with a large number of independent copies of the input superpositions to the original problem. However, in order to achieve this goal, the self-reduction process needs to act on several independent superpositions simultaneously by the same group element. The group action α^t captures this notion. This notion will be crucial for our induction arguments. Also note that the stabilizer and the translating coset, defined later, are the same for group actions α and α^t.

The *stabilizer* of a state $|\phi\rangle \in \Gamma$ is the subgroup $G_{|\phi\rangle} = \{x \in G : |x \cdot \phi\rangle = |\phi\rangle\}$. Given $|\phi\rangle \in \Gamma$, the problem STABILIZER(G, α, t) consists of finding $\mathrm{O}(\log|G|)$ generators for the subgroup $G_{|\phi\rangle}$, given t copies of $|\phi\rangle$.

PROPOSITION 2.1. *Let G be a finite abelian group given as a black-box group with encoding length ℓ and let α be a group action of G. When $t = \Omega(\log(|G|)\log(1/\varepsilon))$, then* STABILIZER$(G, \alpha, t)$ *can be solved in quantum time* $\mathrm{poly}(\ell)\log(1/\varepsilon)$ *with error ε.*

Proof. Let $|\phi\rangle^{\otimes t}$ be the input of STABILIZER. Let f be the quantum function on G defined by $|f(x)\rangle = |x \cdot \phi\rangle$ for every $x \in G$. Observe that f is an instance of the natural extension of HIDDEN SUBGROUP to quantum functions and it hides the stabilizer $G_{|\phi\rangle}$.

The algorithm for STABILIZER is simply the standard algorithm for the abelian HIDDEN SUBGROUP with error ε. In the standard algorithm, every query is of the form $|x\rangle_G|0\rangle_S$. We simulate the ith query $|x\rangle_G|0\rangle_S$ using the ith copy of $|\phi\rangle$. The second register of the query is swapped with $|\phi\rangle$, and then we let x act on it. We remark that the standard algorithm for abelian HIDDEN SUBGROUP outputs $\mathrm{O}(\log|G|)$ generators for the hidden subgroup. $\square$

Note that in general the input superposition $|\phi\rangle^{\otimes t}$ gets destroyed by the above algorithm.

The *orbit* of a state $|\phi\rangle \in \Gamma$ is the subset $G(|\phi\rangle) = \{|x \cdot \phi\rangle : x \in G\}$. Define $|G \cdot \varphi\rangle = \frac{1}{\sqrt{|G(|\phi\rangle)|}} \sum_{|\varphi'\rangle \in G(|\varphi\rangle)} |\varphi'\rangle$. Equivalently, $|G \cdot \phi\rangle = \frac{1}{\sqrt{|G|}} \sum_{x \in G} |x \cdot \phi\rangle$. The *translating coset* of two states $|\phi_0\rangle$ and $|\phi_1\rangle$ of Γ is the set $\{u \in G : |u \cdot \phi_1\rangle = |\phi_0\rangle\}$. The translating coset of $|\phi_0\rangle$ and $|\phi_1\rangle$ is either empty or a left coset $uG_{|\phi_1\rangle}$ (or equivalently a right coset $G_{|\phi_0\rangle}u$) for some $u \in G$. If the latter case occurs, $|\phi_0\rangle$ and $|\phi_1\rangle$ have conjugate stabilizers: $G_{|\phi_0\rangle} = uG_{|\phi_1\rangle}u^{-1}$. TRANSLATING COSET is a generalization of STABILIZER:

TRANSLATING COSET(G, α, t)
Input: t copies of two quantum states $|\phi_0\rangle, |\phi_1\rangle \in \Gamma$.
Output:

- reject if $G(|\phi_0\rangle) \cap G(|\phi_1\rangle) = \emptyset$;
- $u \in G$ such that $|u \cdot \phi_1\rangle = |\phi_0\rangle$ and $\mathrm{O}(\log|G|)$ generators for $G_{|\phi_1\rangle}$ otherwise.

For a function f on G, define the superposition $|f\rangle = \frac{1}{\sqrt{|G|}} \sum_{g \in G} |g\rangle|f(g)\rangle$, and for $x \in G$, define the function $x \cdot f : g \mapsto f(gx)$. Let $\Gamma(f) = \{|x \cdot f\rangle : x \in G\}$. Then a group element x acts naturally on $|f'\rangle \in \Gamma(f)$ by mapping it to the superposition $|x \cdot f'\rangle$. We call this group action the *translation action*. The mapping $|x\rangle|f'\rangle \mapsto |x\rangle|x \cdot f'\rangle$ is realized by right multiplying the first register of $|f'\rangle$ by x^{-1}.

PROPOSITION 2.2. *Suppose G is a finite group and let $t = \mathrm{poly}(\log|G|)$. Then* HIDDEN SUBGROUP$(G)$ *(resp.,* HIDDEN TRANSLATION(G)*) can be solved with a call* *to* STABILIZER(G, τ, t) *(resp.,* TRANSLATING COSET(G, τ, t)*), where* τ *denotes the translation action.*

Proof. Let f be an instance of HIDDEN SUBGROUP. Then the stabilizer of $|f\rangle$ is the group hidden by f. Let (f_0, f_1) be an instance of HIDDEN TRANSLATION. Then the translating coset of $|f_0\rangle$ and $|f_1\rangle$ is a singleton whose element is the translation of (f_0, f_1). $\square$

3. Hidden translation in $\mathbb{Z}_p^n$. In this section, we show that HIDDEN TRANSLATION(G) can be solved in quantum polynomial time in the special case when $G = \mathbb{Z}_p^n$ for any fixed prime number $p > 2$. In this section we use the additive notation for the group operation, and $x \cdot y \in \mathbb{Z}_p$ stands for the standard inner product for $x, y \in \mathbb{Z}_p^n$. Since $\mathbb{Z}_2^n \rtimes Z_2$ is isomorphic to the abelian group $\mathbb{Z}_2^n \times Z_2$, one already has a quantum polynomial time algorithm for HIDDEN TRANSLATION in $\mathbb{Z}_2^n$ by reducing it to HIDDEN SUBGROUP in $\mathbb{Z}_2^{n+1}$ by the method of [13].

For the convenience of the reader we present our method using intermediate measurements. However, the measurements can always be eliminated (see [2]), giving a unitary and therefore reversible algorithm, possibly with errors.

The quantum part of our algorithm consists of performing `FourierSampling` over the abelian group $\mathbb{Z}_p^n \times \mathbb{Z}_2$. It turns out that from the samples we will only use elements of the form $(y, 1)$. The important property of these elements y is that they are *not* orthogonal to the hidden translation. Some properties of the distribution of the samples are stated for general abelian groups in the following lemma.

LEMMA 3.1. *Let G be a finite abelian group. Let $f = (f_0, f_1)$, $f : G \times \mathbb{Z}_2 \to S$ be an instance of* HIDDEN TRANSLATION(G) *having a translation $u \neq 0$. Then* $\texttt{FourierSampling}^f(G \times \mathbb{Z}_2)$ *outputs an element in $G \times \{1\}$ with probability $1/2$. Moreover, the probability of sampling the element $(y, 1)$ depends only on $\chi_y(u)$, and is 0 if and only if $y \in u^\perp$.*

Proof. The state vector of $\texttt{FourierSampling}^f(G \times \mathbb{Z}_2)$ before the final observation is

$$\frac{1}{2|G|} \sum_{x \in G} \sum_{y \in G} \sum_{c=0,1} \chi_y(x) \big(1 + (-1)^c \chi_y(u)\big) |y\rangle |c\rangle |f_0(x)\rangle.$$

Therefore the probability of sampling $(y, 1)$ is proportional to $|1 - \chi_y(u)|^2$, whence the statement follows as $\chi_y(u) = 1$ if and only if $y \in u^\perp$ and $\sum_{y \in G} |1 - \chi_y(u)|^2 = 2|G| - 2\sum_{y \in G} \chi_y(u) = 2|G|$. $\square$

When $G = \mathbb{Z}_p^n$, the value $\chi_y(u) = e^{\frac{2\pi i}{p} y \cdot u}$ depends only on the inner product $y \cdot u$ over $\mathbb{Z}_p$, and $y \in u^\perp$ exactly when $y \cdot u = 0$. Therefore every $(y, 1)$ generated satisfies $y \cdot u \neq 0$. Thus the output distribution is different from the usual one obtained for the abelian HIDDEN SUBGROUP where only vectors orthogonal to the hidden subgroup are generated. We overcome the main obstacle, which is that we do not know the actual value of the inner product $y \cdot u$, by raising these inequations to the power $(p - 1)$. They become a system of polynomial equations of degree at most $(p - 1)$ since $a^{p-1} = 1$ for every nonzero $a \in \mathbb{Z}_p$. In general, solving systems of polynomial equations over any finite field is NP-complete. But using the other special feature of our distribution, which is that the probability of sampling $(y, 1)$ depends only on the inner product $y \cdot u$, we are able to show that—for fixed prime p—after a polynomial number of samplings, our system of equations has a unique solution with constant probability, and the solution can be found in deterministic polynomial time.

To solve our system of polynomial equations, we linearize it in the $(p-1)$th symmetric power of $\mathbb{Z}_p^n$. We think of $\mathbb{Z}_p^n$ as an n-dimensional vector space over $\mathbb{Z}_p$. For a prime number p and an integer $k \geq 0$, let $\mathbb{Z}_p^{(k)}[x_1, \ldots, x_n]$ be the kth symmetric power of $\mathbb{Z}_p^n$ which will be thought of as the vector space, over the finite field $\mathbb{Z}_p$, of homogeneous polynomials of degree k in variables $x_1, \ldots, x_n$. The monomials of degree $(p-1)$ form a basis of $\mathbb{Z}_p^{(p-1)}[x_1, \ldots, x_n]$, whose dimension is therefore $\binom{n+p-2}{p-1}$, which is polynomial in n when p is constant. $\mathbb{Z}_p^{(1)}[x_1, \ldots, x_n]$ is isomorphic to $\mathbb{Z}_p^n$ as a vector space. For two vectors $Y_1, Y_2 \in \mathbb{Z}_p^{(p-1)}[x_1, \ldots, x_n]$, we denote their standard inner product over the monomial basis by $Y_1 \cdot Y_2$.

For every $y = (a_1, \ldots, a_n) \in \mathbb{Z}_p^n$ and positive integer k, we define $y^{(k)} \in \mathbb{Z}_p^{(k)}[x_1, \ldots, x_n]$ as the polynomial $(\sum_{j=1}^n a_j x_j)^k$. For $y = (a_1, \ldots, a_n)$, $z = (b_1, \ldots, b_n)$ in $\mathbb{Z}_p^n$, and positive integers k, l, we define the product $y^{(k)} z^{(l)} \in \mathbb{Z}_p^{(k+l)}[x_1, \ldots, x_n]$ as the polynomial $(\sum_{i=1}^n a_i x_i)^k (\sum_{j=1}^n b_j x_j)^l$. Now observe that if $u = (u_1, \ldots, u_n)$ is the hidden translation vector, then the vector $u^* \in \mathbb{Z}_p^{(k)}[x_1, \ldots, x_n]$ which for every monomial $x_1^{e_1} \cdots x_n^{e_n}$ has coordinate $u_1^{e_1} \cdots u_n^{e_n}$ satisfies $y^{(p-1)} \cdot u^* = (y \cdot u)^{p-1}$. Therefore each linear inequation $y \cdot u \neq 0$ over $\mathbb{Z}_p^n$ will be transformed into the linear equation $y^{(p-1)} \cdot U = 1$ over $\mathbb{Z}_p^{(p-1)}[x_1, \ldots, x_n]$, where U is a $\dim \mathbb{Z}_p^{(p-1)}[x_1, \ldots, x_n]$-sized vector of unknowns.

We will see below that the vectors $y^{(p-1)}$ span the space $\mathbb{Z}_p^{(p-1)}[x_1, \ldots, x_n]$ when y ranges over $\mathbb{Z}_p^n$. Moreover, in what is the main part of our proof, we show in Lemma 3.4 that whenever the span of $y^{(p-1)}$ for the samples y is not $\mathbb{Z}_p^{(p-1)}[x_1, \ldots, x_n]$, our sampling process furnishes with probability at least $1/p$ a vector $z \in \mathbb{Z}_p^n$ such that $z^{(p-1)}$ is linearly independent from the $y^{(p-1)}$ for the previously sampled y's. This immediately implies that if our sample size is of the order of the dimension of $\mathbb{Z}_p^{(p-1)}[x_1, \ldots, x_n]$, the span of $y^{(p-1)}$ for the samples y is $\mathbb{Z}_p^{(p-1)}[x_1, \ldots, x_n]$ with high probability. In that case, the linear equations $y^{(p-1)} \cdot U = 1$ have exactly one solution, which is u^*. From this unique solution one can easily recover a vector v such that $v = au$ for some $0 < a < p$ (note that $v^* = u^*$). Now u can be found by checking the $(p-1)$ possibilities.

The following combinatorial lemma is at the basis of the correctness of our procedure.

Lemma 3.2 (line lemma). *Let $y, z \in \mathbb{Z}_p^n$. For $1 \leq k \leq p-1$, define $L_{z,y}^{(k)} = \{(z + ay)^{(k)} : 0 \leq a \leq k\}$. Then for all $0 \leq l \leq k$, $z^{(l)} y^{(k-l)} \in \operatorname{Span}(L_{z,y}^{(k)})$, where the span is taken with $\mathbb{Z}_p$-coefficients.*

Proof. Let $M_{z,y}^{(k)} = \{z^{(l)} y^{(k-l)} : 0 \leq l \leq k\}$. Clearly, $\operatorname{Span}(L_{z,y}^{(k)}) \subseteq \operatorname{Span}(M_{z,y}^{(k)})$. We claim that the inverse inclusion is also true since the determinant of $L_{z,y}^{(k)}$ in $M_{z,y}^{(k)}$ is nonzero in $\mathbb{Z}_p$. Indeed, it is $\left(\prod_{l=0}^k \binom{k}{l}\right) V(0, 1, \ldots, k)$, where V denotes the Vandermonde determinant. □

Proposition 3.3. *For $1 \leq k \leq p-1$, $\mathbb{Z}_p^{(k)}[x_1, \ldots, x_n]$ is spanned by $y^{(k)}$ as y ranges over $\mathbb{Z}_p^n$.*

Proof. We prove the proposition by induction on k. The base case $k = 1$ is trivial. Suppose the statement holds for k, $1 \leq k < p-1$. Consider a monomial M in $x_1, \ldots, x_n$ of degree $k+1$. If $M = x_i^{k+1}$ for some $1 \leq i \leq n$, then M trivially lies in the span of $y^{(k+1)}$ as y ranges over $\mathbb{Z}_p^n$. Else, $M = x_i M'$ for some $1 \leq i \leq n$ and degree k monomial M'. Let $z \in \mathbb{Z}_p^n$. From Lemma 3.2, we see that $x_i z^{(k)} \in \operatorname{Span}(\{(x_i + az)^{(k+1)} : 0 \leq a \leq k+1\})$. By induction hypothesis, M' lies in the span of $z^{(k)}$ as z ranges over $\mathbb{Z}_p^n$. Hence, $x_i M'$ lies in the span of $x_i z^{(k)}$ as z ranges over

$\mathbb{Z}_p^n$. Thus, $M \in \mathrm{Span}(\{(x_i + az)^{(k+1)} : 0 \leq a \leq k+1, z \in \mathbb{Z}_p^n\})$. This shows that $\mathbb{Z}_p^{(k+1)}[x_1, \ldots, x_n]$ is spanned by $y^{(k+1)}$ as y ranges over $\mathbb{Z}_p^n$, completing the proof of the induction step and also that of the proposition. □

We are now ready to prove the main lemma.

Lemma 3.4. *Let $u \in \mathbb{Z}_p^n$, $u \neq 0$ and let W be a subspace of $\mathbb{Z}_p^{(p-1)}[x_1, \ldots, x_n]$. We set $R = \{y \in \mathbb{Z}_p^n : y^{(p-1)} \in W\}$. For $k = 0, \ldots, p-1$, let $V_k = \{y \in \mathbb{Z}_p^n : y \cdot u = k\}$ and $R_k = R \cap V_k$. If $W \neq \mathbb{Z}_p^{(p-1)}[x_1, \ldots, x_n]$, then $|R_k|/|V_k| \leq (p-1)/p$ for $k = 1, \ldots, p-1$.*

Proof. Observe that $R_k = \{ky : y \in R_1\}$ for $0 < k < p$. Therefore the sets R_k, $0 < k < p$, have the same size. Observe also that the sets V_k, $0 \leq k < p$, have the same size, and they partition $\mathbb{Z}_p^n$. Hence the values $|R_k|/|V_k|$ are the same for $0 < k < p$.

Since $W \neq \mathbb{Z}_p^{(p-1)}[x_1, \ldots, x_n]$, Proposition 3.3 implies that $R \neq \mathbb{Z}_p^n$. We consider two cases. In the first case, $V_0 \subseteq R$. This implies that R_1 is a proper subset of V_1. Choose any $y \in V_1 \setminus R_1$. Then, by Lemma 3.2, in every coset of $\langle y \rangle$ there is an element outside of R. A coset of $\langle y \rangle$ contains exactly one element from each V_k, $k = 0, \ldots, p-1$. Hence $\cup_{k \neq 0} V_k$ is partitioned into equal parts, each part of size $(p-1)$, by intersecting with the cosets of $\langle y \rangle$. In each part, there is an element outside of R. Therefore $|\cup_{k \neq 0} R_k|/|\cup_{k \neq 0} V_k| \leq (p-2)/(p-1)$. Hence, $|R_k|/|V_k| \leq (p-2)/(p-1) < (p-1)/p$ for $k = 1, \ldots, p-1$, and the statement follows.

In the second case, $V_0 \not\subseteq R$. Therefore, there is an element $y \in V_0 \setminus R_0$. Then every V_k, $k = 0, \ldots, p-1$, is a union of cosets of $\langle y \rangle$. Lemma 3.2 implies that every coset of $\langle y \rangle$ contains an element outside of R. This proves that $|R_k|/|V_k| \leq (p-1)/p$ for $k = 0, \ldots, p-1$. This completes the proof of the lemma. □

We now specify the algorithm `TranslationFinding` and prove that, with high probability, it finds the hidden translation in quantum polynomial time when p is constant.

```
TranslationFinding$^f(\mathbb{Z}_p^n)$
  0. If $f_0(0) = f_1(0)$ then output 0.
  1. $N \leftarrow 13p\binom{n+p-2}{p-1}$.
  2. For $i = 1, \ldots, N$ do
         $(z_i, b_i) \leftarrow$ FourierSampling$^f(\mathbb{Z}_p^n \times \mathbb{Z}_2)$.
  3. $\{y_1, \ldots, y_M\} \leftarrow \{z_i : b_i = 1\}$.
  4. For $i = 1, \ldots, M$ do $Y_i \leftarrow y_i^{(p-1)}$.
  5. Solve the system of linear equations
         $Y_1 \cdot U = 1, \ldots, Y_M \cdot U = 1$.
  6. If there are no solutions or more than one solution, then abort.
  7. Let $1 \leq j \leq n$ be such that the coefficient of $x_j^{p-1}$ is 1 in $U$.
  8. Let $v = (v_1, \ldots, v_n) \in \mathbb{Z}_p^n$ be such that $v_j = 1$ and $v_k$ is the
     coordinate of $x_k x_j^{p-2}$ in $U$ for $k \neq j$.
  9. Find $0 < a < p$ such that $f_0(0) = f_1(av)$.
 10. Output $av$.
```

Theorem 3.5. *For every prime number p, every integer $n \geq 1$, and every function $f : \mathbb{Z}_p^n \times Z_2 \to S$ having a translation given via a quantum oracle, algorithm* `TranslationFinding`$^f(\mathbb{Z}_p^n)$ *aborts with probability less than $1/2$, and when it does not abort it outputs the translation of f. The query complexity of the algorithm is $\mathrm{O}(p(n+p)^{p-1})$, and its time complexity is $(n+p)^{\mathrm{O}(p)}$.*

Proof. Because of Step 0 of the algorithm, we can suppose without loss of generality (w.l.o.g.) that the translation u of f is nonzero.

If the algorithm does not abort, then $U = u^*$ is the unique solution of the system in step 5. When the coefficient of x_j^{p-1} is 1 in U, then $u_j \neq 0$. Also, $u_k = u_j v_k$ for every k. Thus, $u = u_j v$ and u is found in step 9 for $a = u_j$.

From Lemma 3.1, we see that the probability that the algorithm `Fourier Sampling`$^f(\mathbb{Z}_p^n \times \mathbb{Z}_2)$ outputs $(y, 1)$ for some y is $1/2$. Therefore the expected value of M is $N/2$, and $M < N/3$ with probability at most $e^{-N/18} < 1/4$ because of Chernoff's bound. If the system $Y_1, \ldots, Y_M$ has full rank, then it has a unique solution. By Lemmas 3.1 and 3.4, the expected number of linear equations that guarantee that the system has full rank is at most $p\binom{n+p-2}{p-1}$. Since $N/3 > 4p\binom{n+p-2}{p-1}$, by Markov's inequality, the solution U is unique with probability at least $3/4$. Thus, the total probability of aborting is less than $1/2$. □

COROLLARY 3.6. *Let p be a prime. Then the problem of* HIDDEN TRANSLATION$(\mathbb{Z}_p^n)$ *can be solved in quantum time $(n+p)^{\mathrm{O}(p)}\log(1/\varepsilon)$ with error ε using $t = \Theta(p(n+p)^{p-1}\log(1/\varepsilon))$ accesses to the oracles for f_0, f_1.*

Proof. We perform two modifications in the algorithm `TranslationFinding`. First, to get error ε, the integer N is multiplied by $\mathrm{O}(\log(1/\varepsilon))$. Moreover, we assumed in the algorithm that there is an oracle for $f = (f_0, f_1)$, which was used to choose f_b knowing b. This is not possible in general when f_0 and f_1 are given by two distinct oracles. Therefore we replace the oracle access $|x\rangle|b\rangle|0\rangle_S \mapsto |x\rangle|b\rangle|f_b(x)\rangle_S$ by

$$|x\rangle|b\rangle|0\rangle_S|0\rangle_S \mapsto |x\rangle|b\rangle|f_b(x)\rangle_S|f_{1-b}(-x)\rangle_S.$$

This type of quantum oracle corresponds to the function $f' = (f'_0, f'_1)$, where $f'_0(x) = (f_0(x), f_1(x))$ and $f'_1(x) = (f_1(x), f_0(-x))$. Obviously, f'_0 is injective and $f'_0(x) = f'_1(x+u)$. We can apply Theorem 3.5 in this new setting.

Let us now show how to simulate this new oracle access. From $|x\rangle|b\rangle|0\rangle_S|0\rangle_S$ we compute $|(-1)^b x\rangle|b\rangle|0\rangle_S|0\rangle_S$, and then we call f_0 and get $|(-1)^b x\rangle|b\rangle|f_0((-1)^b x)\rangle_S|0\rangle_S$. We multiply the first register by (-1) and call f_1, which gives

$$|(-1)^{b+1}x\rangle|b\rangle|f_0((-1)^b x)\rangle_S|f_1((-1)^{b+1}x)\rangle_S.$$

Finally, we multiply the first register by $(-1)^{b+1}$ and swap the last two registers when $b = 1$. □

As there is a quantum reduction from HIDDEN SUBGROUP in $\mathbb{Z}_p^n \rtimes \mathbb{Z}_2$ to HIDDEN TRANSLATION in $\mathbb{Z}_p^n$ by the method of [13], we obtain the following corollary.

COROLLARY 3.7. *Let p be a fixed prime. Then* HIDDEN SUBGROUP$(\mathbb{Z}_p^n \rtimes \mathbb{Z}_2)$ *can be solved in quantum time* poly(n).

The algorithm `TranslationFinding` can also be extended to solve TRANSLATING COSET in $\mathbb{Z}_p^n$.

COROLLARY 3.8. *Let p be a prime. Let α be a group action of $\mathbb{Z}_p^n$. When $t = \Omega(p(n+p)^{p-1}\log(1/\varepsilon))$,* TRANSLATING COSET$(\mathbb{Z}_p^n, \alpha, t)$ *can be solved in quantum time $(n+p)^{\mathrm{O}(p)}\log(1/\varepsilon)$ with error ε.*

Proof. Let the input of the TRANSLATING COSET$(\mathbb{Z}_p^n, \alpha, t)$ be $(|\phi_0\rangle^{\otimes t}, |\phi_1\rangle^{\otimes t})$. We can suppose w.l.o.g. that the stabilizers of $|\phi_0\rangle$ and $|\phi_1\rangle$ are trivial. Indeed the stabilizers can be computed by Proposition 2.1. If they are different, then the algorithm obviously has to reject; otherwise we work in the factor group $\mathbb{Z}_p^n/G_{|\phi_0\rangle} \cong \mathbb{Z}_p^{n'}$ for some $n' \leq n$. To be more specific, we can compute a ($\mathbb{Z}_p$-basis for) a subgroup G_1 of $\mathbb{Z}_p^n$ which is a direct complement of $G_{|\phi_0\rangle}$ by augmenting a basis for $G_{|\phi_0\rangle}$ to a basis of $\mathbb{Z}_p^n$, and we can actually work with G_1 in place of G.

For $b = 0, 1$, let f_b be the injective quantum function on G defined by $|f_b(x)\rangle = |x \cdot \phi_b\rangle$ for every $x \in G$. If the translating coset of $(|\phi_0\rangle, |\phi_1\rangle)$ is empty, then f_0 and

f_1 have distinct ranges. Otherwise the translating coset of $(|\phi_0\rangle, |\phi_1\rangle)$ is a singleton $\{u\}$, and (f_0, f_1) have the translation u.

The algorithm for TRANSLATING COSET on input $(|\phi_0\rangle^{\otimes t}, |\phi_1\rangle^{\otimes t})$ is the algorithm `TranslationFinding` on input (f_0, f_1) with a few modifications described below. The oracle access to (f_0, f_1) is modified in the same way as in Corollary 3.6. We simulate the ith query $|x\rangle|b\rangle|0\rangle_S|0\rangle_S$ using the ith copy of $|\phi_0\rangle|\phi_1\rangle$. The two registers $|0\rangle_S|0\rangle_S$ are swapped with $|\phi_b\rangle|\phi_{1-b}\rangle$, and then we let act x on $|\phi_b\rangle$ and $(-x)$ on $|\phi_{1-b}\rangle$.

The equality tests in steps 0 and 9 are replaced by the swap test [7, 18] iterated $\mathrm{O}(\log(1/\varepsilon))$ times. Finally, N is multiplied by $\mathrm{O}(\log(1/\varepsilon))$, and the algorithm rejects whenever the algorithm `TranslationFinding` aborts or there is no solution in step 9. □

4. Translating coset in solvable groups.

4.1. Preparation.

4.1.1. Quantization of the problems. Let G be a black-box group with unique encoding, and let α be a group action on Γ.

We now describe quantum analogues of problems with classical outcomes, as unitary transformations whose outputs are basically uniform superpositions on the possible classical outcomes.

We will give quantum circuit implementations for the new problems. A quantum circuit has both *input/output* registers and *ancilla* registers. The latter ones are initialized to some default value, usually a 0-string, that we denote $|0\rangle$. We will explicitly mention when we consider a different default value. We identify a quantum circuit with the unitary transformation it defines.

Let U be a unitary transformation. A quantum circuit C *implements* U if $C = U \otimes \mathrm{Id}$, where the tensor product is between input/output registers and ancilla registers. Most often, our unitary transformations will be only partially specified, and our quantum circuits will only approximately implement them. This motivates the following generalization of implementation.

A *partial unitary* U is a transformation defined on a subset $\mathcal{S}$ of a Hilbert space H, such that there exists a unitary transformation V on H which coincides with U on $\mathcal{S}$. A quantum circuit C *implements* U *on* $\mathcal{S}$ *with error* ε if $C(|\psi\rangle \otimes |0\rangle)$ and $U|\psi\rangle \otimes |0\rangle$ are of trace distance at most ε for every $|\psi\rangle \in \mathcal{S}$. We will omit ε when $\varepsilon = 0$, and also $\mathcal{S}$ when it is understood from the context.

Given a circuit that implements a unitary U, one can design a circuit of the same size that implements the unitary U^{-1} by applying backward the circuit for U, where each gate is replaced by its inverse. Therefore in our model, the complexity for implementing a unitary transformation or its inverse is the same. Thus we will say that a circuit uses as *black boxes t implementations of* U whenever it uses t gates U or U^{-1}.

Our notion of implementation does not allow any garbage in the computation: at the end of the computation the ancilla registers must come back to their initial default value, potentially approximately. It is not always true for a quantum circuit, even if it computes the desired outcomes. In that case we will say that the computation is *with garbage*. Nonetheless, when a quantum circuit computes a classical function without error, we can assume that the computation is without garbage using the standard cleaning method: run the circuit C, XOR the output in a new register (initialized to the 0-string), undo the circuit by running C^{-1}. In such a situation, we will therefore always assume that we have at our disposal such a circuit without garbage.

DEFINITION 4.1. *Let* $g_1, g_2, \ldots, g_k \in G$ *and* $H = \langle g_1, g_2, \ldots, g_k\rangle$. SUBGROUP

Superposition$(G, g_1, g_2, \ldots, g_k)$ *is a partial unitary transformation that maps state* $|1_G\rangle$ *to state* $|H\rangle$.

In the following description of a quantum circuit, we write in boldface the input registers of the circuit, whereas fresh registers are in regular font. The output registers are a priori the same as the input registers. We also assume for simplicity that we have at our disposal a zero-error quantum algorithm for computing the generalized discrete logarithm and for order finding. The actual implementations for the single basis element case [41] and for the general case [26] introduce only exponentially small errors. Note that one can also use a generalization for the single basis element case of [34] which is without error. We also note that the heart of the circuit is not to compute H from the generators, but rather to create the superposition over H by uncomputing the discrete log.

```
AbelianGS$(G, g_1, g_2, \ldots, g_k)$
Hypothesis:  $H = \langle g_1, g_2, \ldots, g_l \rangle$ is abelian.
Input:  $|\mathbf{1}_{\boldsymbol{G}}\rangle$
  1. Compute a basis $h_1, h_2, \ldots, h_l$ such that $\langle h_1\rangle \times \langle h_2\rangle \times \cdots \times \langle h_l\rangle = H$, and
     the respective orders $r_j$ of $h_j$.
  2. Compute in a fresh register the superposition
         $\sum_{0 \le a_j < r_j} |a_1, a_2, \ldots, a_l\rangle |\mathbf{1}_{\boldsymbol{G}}\rangle.$
  3. Perform fast exponentiation $h_j^{a_j}$ in fresh register:
         $\sum_{0 \le a_j < r_j} |a_1, a_2, \ldots, a_l\rangle |h_1^{a_1}, h_2^{a_2}, \ldots, h_l^{a_l}\rangle |\mathbf{1}_{\boldsymbol{G}}\rangle.$
  4. Multiply $1_G$ by all the $h_j^{a_j}$:
         $\sum_{0 \le a_j < r_j} |a_1, a_2, \ldots, a_l\rangle |h_1^{a_1}, h_2^{a_2}, \ldots, h_l^{a_l}\rangle |\boldsymbol{h_1^{a_1} h_2^{a_2} \ldots h_l^{a_l}}\rangle.$
  5. Undo step 3.
         $\sum_{0 \le a_j < r_j} |a_1, a_2, \ldots, a_l\rangle |\boldsymbol{h_1^{a_1} h_2^{a_2} \ldots h_l^{a_l}}\rangle.$
  6. Undo the computation of the generalized discrete logarithm of the
     group elements $h_1^{a_1} h_2^{a_2} \ldots h_l^{a_l}$ in the basis $(h_1, h_2, \ldots, h_l)$:
         $\sum_{0 \le a_j < r_j} |\boldsymbol{h_1^{a_1} h_2^{a_2} \ldots h_l^{a_l}}\rangle = |\boldsymbol{H}\rangle.$
  7. Undo step 1.
```

Theorem 4.2. *Let G be a black-box group with unique encoding of length ℓ. Let $g_1, g_2, \ldots, g_k \in G$ be generators of an abelian subgroup H. Then* AbelianGS$(G, g_1, g_2, \ldots, g_k)$ *implements* Subgroup Superposition$(G, g_1, g_2, \ldots, g_k)$ *in quantum time* poly$(k\ell)$.

Proof. Since the description of the algorithm is clear, the proof consists in checking that all the tasks involved in AbelianGS$(G, g_1, g_2, \ldots, g_k)$ can be done in the requested complexity.

The main potential difficulty is for step 1. This step can be done in quantum time poly$(k\ell)$ using the method of [9] without error since we assume that we can do quantum Fourier transform without error on abelian groups. □

For solvable groups, we consider the following extension, which produces the required superposition, but with garbage.

THEOREM 4.3 (see [46]). *Let G be a black-box group with unique encoding of length ℓ. Given generators $g_1, g_2, \ldots, g_k \in G$ of a solvable subgroup H, the state $|H\rangle$ can be produced with error ε and with garbage in quantum time* $\mathrm{poly}(k\ell)\log(1/\varepsilon)$.

Now we define the quantized versions of TRANSLATING COSET and STABILIZER. (These are descriptions of certain unitary transformations.) Recall that if T is empty, then $|T\rangle = |\emptyset\rangle$, where $|\emptyset\rangle$ is a specific basis element.

DEFINITION 4.4. TRANSLATING COSET SUPERPOSITION(G, α, t) *is the partial unitary transformation that maps state $|\phi_0\rangle^{\otimes t}|\phi_1\rangle^{\otimes t}|1_G\rangle$ to state $|\phi_0\rangle^{\otimes t}|\phi_1\rangle^{\otimes t}|T\rangle$, where $T = \{u \in G : |u \cdot \phi_1\rangle = |\phi_0\rangle\}$.* STABILIZER SUPERPOSITION *is the special case of* TRANSLATING COSET SUPERPOSITION *with $|\phi_1\rangle = |\phi_0\rangle$.*

In general $O(\log|G| \log \frac{1}{\varepsilon})$ copies of the coset superposition $|T\rangle$ are sufficient to determine T classically with error probability ε. To see this, assume that we have state $|T\rangle^{\otimes s}$. We then multiply the contents of the second, third, etc., register by the inverse of the group element in the first register. Then the first register will contain an element representing the coset, while in the remaining register there are elements of the stabilizer subgroup which, if s is large enough, will contain a system of generators with high probability.

```
ElementaryAbelianTCS($G, \alpha, t$)
Hypothesis:  $G \cong \mathbb{Z}_p^n$
Input:  $|\boldsymbol{\phi_0}\rangle^{\otimes t}|\boldsymbol{\phi_1}\rangle^{\otimes t}|\mathbf{1_G}\rangle$
  1. Apply the algorithm of Corollary 3.8 on the first $2t$ input
     registers, using a fresh register for the computation:

       $$\sum_{u\in G, X\in G^{\le \log|G|}} \alpha_{u,X}|u,X\rangle|\theta_{u,X}\rangle|\mathbf{1_G}\rangle,$$

     where $|u,X\rangle$ denotes the output of the algorithm of Corollary 3.8,
     and $|\theta_{u,X}\rangle$ denotes the other remaining registers.[2]
  2. Apply AbelianGS($G, X$) to the last input register:

       $$\sum_{u\in G, X\in G^{\le \log|G|}} \alpha_{u,X}|u,X\rangle|\theta_{u,X}\rangle|\langle\boldsymbol{X}\rangle\rangle.$$

  3. Left multiply the last input register by $u$:

       $$\sum_{u\in G, X\in G^{\le \log|G|}} \alpha_{u,X}|u,X\rangle|\theta_{u,X}\rangle|\boldsymbol{u}\langle\boldsymbol{X}\rangle\rangle.$$

  4. Undo step 1.
```

COROLLARY 4.5. *Let $G \cong \mathbb{Z}_p^n$ be a black-box group with unique encoding of length ℓ. Let α be a group action of G and let $t = \Omega(p(n+p)^{p-1}\log(1/\varepsilon))$ be a positive integer. Then* ElementaryAbelianTCS(G, α, t) *implements* TRANSLATING COSET SUPERPOSITION(G, α, t) *with error ε in quantum time* $\ell^{\mathrm{O}(1)}(n+p)^{\mathrm{O}(p)}\log(1/\varepsilon)$.

Proof. In the first step of the algorithm, X denotes a set of generators for $G_{|\phi_1\rangle}$ and u a group element such that $|u \cdot \phi_1\rangle = |\phi_0\rangle$. When no solution exists, we simply

[2] The sum is over all elements $u \in G$ and all lists X consisting of at most $\log|G|$ elements of G. If there were no errors, $\alpha_{u,X}$ would be zero for pairs (u, X) which do not describe the coset translating ϕ_0 to ϕ_1. Due to errors of the algorithm, some such coefficients can be nonzero, although very small.

request the algorithm of Corollary 3.8 to set $X = \emptyset$, and let u be any group element, instead of rejecting.

Let v be a fixed group element such that $|v \cdot \phi_1\rangle = |\phi_0\rangle$. Because of the choice of the parameters and by Corollary 3.8, the states $|u\langle X\rangle\rangle$ and $|vG_{|\phi_1\rangle}\rangle$ are of trace distance at most ε. This implies that the final state of the algorithm is of trace distance at most ε from the following state without garbage: $|\phi_0\rangle^{\otimes t}|\phi_1\rangle^{\otimes t}|vG_{|\phi_1\rangle}\rangle$. □

For an arbitrary abelian group G, we can modify procedure `ElementaryAbelianTCS`(G, α, t) by replacing the algorithm of Corollary 3.8 with an adapted version of Kuperberg's subexponential method (see Theorem 7.1 of [31]) to solve Translating Coset. (We only need modifications to Kuperberg's algorithm like the ones to `TranslationFinding` described in the proof of Corollary 3.8: We use the "conditionally swapped pairs of functions" trick presented in the proof of Corollary 3.6 and simulate the oracle with input quantum states.) Let us call the resulting procedure `ArbitraryAbelianTCS`(G, α, t). We obtain the following.

Corollary 4.6. *Let G be a black-box abelian group with unique encoding of length ℓ. Let α be a group action of G, and let $t = 2^{\Omega(\sqrt{\log|G|})}$ be a positive integer. Then* `ArbitraryAbelianTCS`(G, α, t) *implements* Translating Coset Superposition(G, α, t) *with error ε in quantum time $\ell^{\mathrm{O}(1)} 2^{\mathrm{O}(\sqrt{\log|G|})} \log(1/\varepsilon)$.*

4.1.2. Compatible encodings. We will apply recursion into factor groups of solvable groups. Therefore we need an efficient procedure to design a unique encoding for these factor groups. Moreover, for the purpose of our algorithm we will require this encoding to be compatible with the original encoding of the group in the following sense.

Definition 4.7. *Let G be a black-box group with unique encoding* enc *of length ℓ. Let N be a normal subgroup of G. A unique encoding enc_N for G/N is* compatible *with* enc *if*

1. *for every $x \in G$, there is $y \in xN$ such that $\mathsf{enc}_N(xN) = \mathsf{enc}(y)$;*
2. *the partial unitary $|\mathsf{enc}(x)\rangle|0\rangle \mapsto |\mathsf{enc}(x)\rangle|\mathsf{enc}_N(xN)\rangle$, where $x \in G$, can be implemented in quantum time* $\mathrm{poly}(\ell)$*.*

Note that if G has encoding length ℓ, then a compatible encoding for G/N also has encoding length ℓ.

From now on, we assume for simplicity that we have at our disposal a multiple r of $|G|$ such that $r = \mathrm{O}(\ell)$. This multiple is given or computed once for a group, and we keep the same value for all its subgroups. This assumption is reasonable since for solvable groups the cardinality of G can be computed in time $\mathrm{poly}(\ell)$ [46].

In the following theorem, we assume for simplicity that we have at our disposal a zero-error quantum algorithm for computing the generalized discrete logarithm and for order finding.

Theorem 4.8. *Let G be a black-box solvable group with unique encoding* enc *of length ℓ. Let N be a normal subgroup of G such that G/N is abelian. Assume that $\mathrm{O}(\ell)$ copies of $|N\rangle$ are given. There exists a unique encoding enc_N for G/N such that*

1. *a set of generators for G/N, whose size is at most* $\log|G/N|$*, can be computed in quantum time* $\mathrm{poly}(\ell)$*;*
2. *group operations over G/N using encoding enc_N can be computed in quantum time* $\mathrm{poly}(\ell)$*;*
3. *enc_N is compatible with* enc.

Note that even if all the tasks (1) and (2) will use as ancilla several copies of $|N\rangle$, these copies are always restored at the end of the computations. Indeed, since the outcomes of tasks (1) and (2) are classical, one can XOR their value in a fresh register, and reverse the procedure in order to be garbage free and restore the used copies of $|N\rangle$.

Proof. Let $g_1, g_2, \ldots, g_k \in G$ be the generators defining G, where $k = \mathrm{O}(\ell)$, and let r be a known multiple of $|G|$ such that $\log r = \mathrm{O}(\ell)$. The cosets $g_1N, g_2N, \ldots, g_kN$ are generators of G/N. We now show how to learn the structure of the abelian group G/N, and in particular how to extract a subset of at most $\log|G/N|$ generators.

Following the approach of [26], we consider extensions of the quantum algorithms for computing the generalized discrete logarithm and for order finding to functions having quantum ranges. More precisely, for order finding, the function $a \in \mathbb{Z}_r \mapsto |x^aN\rangle$ hides the subgroup $\mathbb{Z}_{r_x}$, where r_x is the order of xN. This function is encoded by the partial unitary map $|a\rangle|N\rangle \mapsto |a\rangle|x^aN\rangle$, which admits a poly$(\ell)$-size circuit, since it can be implemented using $\mathrm{O}(\log r)$ group operations. Thus the algorithm requires as many copies of $|N\rangle$ as the number of function evaluations, that is, $\mathrm{O}(\ell)$. Similarly, given $x \in G$ and $y \in x^aN$, for some unknown $0 \leq a < |G/N|$, one can compute a using $\mathrm{O}(\ell)$ copies of $|N\rangle$.

More generally, one can learn the structure of G/N as in [9] using $\mathrm{O}(\ell)$ copies of $|N\rangle$ and the unitary

$$|a_1, a_2, \ldots, a_k\rangle|N\rangle \mapsto |a_1, a_2, \ldots, a_k\rangle|g_1^{a_1}g_2^{a_2}\ldots g_k^{a_k}N\rangle,$$

where $a_i \in \mathbb{Z}_r$, and the group elements g_i are implicitly encoded using enc. Given the structure of G/N, we are able to find the lexicographically smallest nonredundant subset of generators for G/N from $g_1N, g_2N, \ldots, g_kN$ by throwing out g_i if it is contained in the subgroup of G generated by $g_1, \ldots, g_{i-1}$ and N. Without loss of generality we can assume that this set is $g_1N, g_2N, \ldots, g_jN$. By nonredundancy, we must have $j \leq \log|G/N|$. This full construction of generators $g_1N, g_2N, \ldots, g_jN$ can be done in quantum time poly(ℓ), and therefore condition (1) is satisfied.

For every $i = 1, 2, \ldots, j$, let l_i be the least positive integer such that $g_i^{l_i} \in \langle N, g_1, g_2, \ldots, g_{i-1}\rangle$. Then we can define our compatible encoding by

$$\mathsf{enc}_N(xN) = \mathsf{enc}(g_1^{a_1}g_2^{a_2}\ldots g_j^{a_j}), \quad \text{where } xN = g_1^{a_1}g_2^{a_2}\ldots g_j^{a_j}N \text{ and } 0 \leq a_i < l_i.$$

Since the exponents a_i are uniquely defined, the encoding is unique and satisfies condition (1) of the definition of compatible encodings (Definition 4.7). In order to satisfy the conditions of compatible encodings, and therefore condition (3) of the theorem, we show how to compute in quantum time poly(ℓ) $\mathsf{enc}_N(xN)$ from $\mathsf{enc}(x)$. Again we follow the approach of [26]. Consider the unitary

$$|b, b_1, b_2, \ldots, b_j\rangle|N\rangle \mapsto |b, b_1, b_2, \ldots, b_j\rangle|x^{-b}g_1^{b_1}g_2^{b_2}\ldots g_j^{b_j}N\rangle.$$

This unitary hides a subgroup H of $\mathbb{Z}_r \times \mathbb{Z}_{l_1} \times \cdots \times \mathbb{Z}_j$ generated by a generator of type $u = (1, a_1, a_2, \ldots, a_j)$, where $xN = g_1^{a_1}g_2^{a_2}\ldots g_j^{a_j}N$. Therefore $\mathsf{enc}_N(xN) = \mathsf{enc}(g_1^{a_1}g_2^{a_2}\ldots g_j^{a_j})$. The subgroup H, and therefore the generator u of this particular form, can be found in quantum time poly(ℓ) since this is the solution of Hidden Subgroup for abelian groups extended to functions having quantum ranges [26].

Finally, condition (2) is easily satisfied. Indeed, by the compatibility of our encoding, group operations over G/N can be simulated by one call to the group oracle for G. Then the result $\mathsf{enc}(x)$, for some $x \in G$, has to be converted to $\mathsf{enc}_N(xN)$ using the above procedure. □

4.2. Orbit superposition. In this section, we show that computing the uniform superposition of the orbit of a given state is reducible to instances of Translating

Coset Superposition. In the following definition, we denote by $|G \cdot \varphi\rangle$ the state $\frac{1}{\sqrt{|G(|\phi\rangle)|}} \sum_{|\phi'\rangle \in G(|\phi\rangle)} |\phi'\rangle^{\otimes s}$, where $|\varphi\rangle = |\phi\rangle^{\otimes s}$.

Definition 4.9. *Let $|\phi\rangle \in \Gamma$. Let s be a positive integer and let $|\varphi\rangle = |\phi\rangle^{\otimes s}$. Orbit Superposition$(G, \alpha, s)$ is the partial unitary transformation that maps state $|\varphi\rangle|\varphi\rangle|G\rangle$ to $|G \cdot \varphi\rangle|\varphi\rangle|1_G\rangle$.*

Then the following algorithm implements Orbit Superposition.

```
OS$(G, \alpha, s)$
Input:  $|\boldsymbol{\phi}\rangle^{\otimes 2s}|\boldsymbol{G}\rangle$
  1. Apply the group element in 3rd register to the first $s$ registers:
        $$\sum_{x \in G} |\boldsymbol{x} \cdot \boldsymbol{\phi}\rangle^{\otimes s}|\boldsymbol{\phi}\rangle^{\otimes s}|\boldsymbol{x}\rangle = \sum_{|\varphi'\rangle \in G(|\varphi\rangle)} |\boldsymbol{\varphi'}\rangle|\boldsymbol{\varphi}\rangle|\boldsymbol{x}\boldsymbol{G}_{|\boldsymbol{\phi}\rangle}\rangle,$$
     where $|\varphi\rangle = |\phi\rangle^{\otimes s}$
  2. Perform the inverse of Translating Coset Superposition$(G, \alpha, s)$
     (which maps $|x \cdot \phi\rangle^{\otimes s}|\phi\rangle^{\otimes s}|1_G\rangle$ to $|x \cdot \phi\rangle^{\otimes s}|\phi\rangle^{\otimes s}|xG_{|\phi\rangle}\rangle$):
        $$|\boldsymbol{G} \cdot \boldsymbol{\varphi}\rangle|\boldsymbol{\varphi}\rangle|\boldsymbol{1_G}\rangle.$$
```

Theorem 4.10. *Let G be a black-box group with unique encoding of length ℓ, and let α be a group action on Γ. Let $|\phi\rangle \in \Gamma$. Let s be a positive integer and $|\varphi\rangle = |\phi\rangle^{\otimes s}$. $\mathtt{OS}(G, \alpha, s)$ implements Orbit Superposition(G, α, s) using as a black box one implementation of Translating Coset Superposition(G, α, s) and quantum time* poly(ℓs) *for the remaining computation.*

4.3. Translating coset self-reducibility in solvable groups.

The purpose of this section is to prove Theorem 4.11 stating the reducibility of Translating Coset in some solvable group G to Translating Coset in proper normal subgroups and factors of G under some conditions. Given a group action α of G on a finite set Γ of mutually orthogonal quantum states, we define for every proper normal subgroup $N \lhd G$ the group action α_N of G/N on $\{|N \cdot \phi\rangle : |\phi\rangle \in \Gamma\}$ by $\alpha_N(xN, |N \cdot \phi\rangle) = |x \cdot (N \cdot \phi)\rangle$ for every $x \in G$ and $|\phi\rangle \in \Gamma$. Note that this action is independent of the chosen coset representative x; it depends only on the coset xN.

For a group action like α^s on Γ^s the group action $(\alpha^s)_N$ will act on states such as $|N \cdot \varphi\rangle = \frac{1}{\sqrt{|N(|\phi\rangle)|}} \sum_{|\phi'\rangle \in N(|\phi\rangle)} |\phi'\rangle^{\otimes s}$, where $|\phi\rangle \in \Gamma$ and $|\varphi\rangle = |\phi\rangle^{\otimes s}$.

Note that the use of our notion of compatible encodings allows us to treat the oracle for α as an oracle for α_N.

In the following algorithm, we implicitly use the encoding enc of G for its elements $z \in G$, and a compatible encoding enc_N for G/N (given by Theorem 4.8) for its cosets $zN \in G/N$. Last, for a subset $S \subseteq G$, the notation S/N is the following subset of G/N: $S/N = \{xN : x \in S\}$. In particular, for any subgroup $H \leq G$, we have $uHN/N = \{uhN : hN \in HN/N\}$.

```
TCS$(G, N, \alpha, s(t+1))$
Hypothesis:  $N \lhd G$ with compatible encoding for $G/N$
Input:  $|\boldsymbol{\phi_0}\rangle^{\otimes s(t+1)}|\boldsymbol{\phi_1}\rangle^{\otimes s(t+1)}|\boldsymbol{1_G}\rangle$
Ancilla:  $|N\rangle^{\otimes 2t}|0\rangle$
  1. Perform $t$ times OS$(N, \alpha, s)$ on blocks $|\phi_0\rangle^{\otimes s}|N\rangle$
     and then $t$ times on blocks $|\phi_1\rangle^{\otimes s}|N\rangle$:
        $$|\boldsymbol{N} \cdot \boldsymbol{\varphi_0}\rangle^{\otimes t}|\boldsymbol{\phi_0}\rangle^{\otimes s}|\boldsymbol{N} \cdot \boldsymbol{\varphi_1}\rangle^{\otimes t}|\boldsymbol{\phi_1}\rangle^{\otimes s}|\boldsymbol{1_G}\rangle|1_G\rangle^{\otimes 2t}|0\rangle,$$
     where $|\varphi_0\rangle = |\phi_0\rangle^{\otimes s}$ and $|\varphi_1\rangle = |\phi_1\rangle^{\otimes s}$.
```

2. XOR the compatible encoding of $1_{G/N}$ into the ancilla register $|0\rangle$:
$$\boldsymbol{|N\cdot\varphi_0\rangle^{\otimes t}|\phi_0\rangle^{\otimes s}|N\cdot\varphi_1\rangle^{\otimes t}|\phi_1\rangle^{\otimes s}|1_G\rangle}|1_G\rangle^{\otimes 2t}|1_{G/N}\rangle.$$
3. Perform Translating Coset Superposition$(G/N,(\alpha^s)_N,t)$ on $|N\cdot\varphi_0\rangle^{\otimes t}|N\cdot\varphi_1\rangle^{\otimes t}|1_{G/N}\rangle$:
$$\boldsymbol{|N\cdot\varphi_0\rangle^{\otimes t}|\phi_0\rangle^{\otimes s}|N\cdot\varphi_1\rangle^{\otimes t}|\phi_1\rangle^{\otimes s}|1_G\rangle}|1_G\rangle^{\otimes 2t}|uHN/N\rangle,$$
where $H=G_{|\phi_1\rangle}$ and $|u\cdot\phi_1\rangle=|\phi_0\rangle$, if there is any;
$$\boldsymbol{|N\cdot\varphi_0\rangle^{\otimes t}|\phi_0\rangle^{\otimes s}|N\cdot\varphi_1\rangle^{\otimes t}|\phi_1\rangle^{\otimes s}|1_G\rangle}|1_G\rangle^{\otimes 2t}|\emptyset\rangle$$
otherwise
4. Undo step 1:
$$\boldsymbol{|\phi_0\rangle^{\otimes s(t+1)}|\phi_1\rangle^{\otimes s(t+1)}|1_G\rangle}|N\rangle^{\otimes 2t}|uHN/N\rangle,$$
or
$$\boldsymbol{|\phi_0\rangle^{\otimes s(t+1)}|\phi_1\rangle^{\otimes s(t+1)}|1_G\rangle}|N\rangle^{\otimes 2t}|\emptyset\rangle.$$
In the second case, Stop the algorithm here.
5. Perform s applications of inverse of group element in the last register to registers $|\phi_0\rangle$ (viewed as an element of G thanks to the compatible encoding of G/N):
$$\sum_{zN\in uHN/N}\boldsymbol{|z^{-1}\cdot\phi_0\rangle^{\otimes s}|\phi_0\rangle^{\otimes st}|\phi_1\rangle^{\otimes s}|\phi_1\rangle^{\otimes st}|1_G\rangle}|N\rangle^{\otimes 2t}|\mathsf{enc}_N(zN)\rangle.^{3}$$
6. Perform Translating Coset Superposition(N,α,s) on $|z^{-1}\cdot\phi_0\rangle^{\otimes s}|\phi_1\rangle^{\otimes s}|1_G\rangle$:
$$\sum_{zN\in uHN/N}\boldsymbol{|z^{-1}\cdot\phi_0\rangle^{\otimes s}|\phi_0\rangle^{\otimes st}|\phi_1\rangle^{\otimes s}|\phi_1\rangle^{\otimes st}|n_z(H\cap N)\rangle}|N\rangle^{\otimes 2t}|\mathsf{enc}_N(zN)\rangle.$$
(See the proof of Theorem 4.11 for notation and justification.)
7. Apply the group element in the last register to the first s registers $|z^{-1}\cdot\phi_0\rangle$:
$$\sum_{zN\in uHN/N}\boldsymbol{|\phi_0\rangle^{\otimes s(t+1)}|\phi_1\rangle^{\otimes s(t+1)}|n_z(H\cap N)\rangle}|N\rangle^{\otimes 2t}|\mathsf{enc}_N(zN)\rangle.$$
8. Left multiply by the group element in the last register the group element in the $2s(t+1)+1$st register
$$\sum_{zN\in uHN/N}\boldsymbol{|\phi_0\rangle^{\otimes s(t+1)}|\phi_1\rangle^{\otimes s(t+1)}|zn_z(H\cap N)\rangle}|N\rangle^{\otimes 2t}|\mathsf{enc}_N(zN)\rangle.$$
9. Inverse in the last and the $2s(t+1)+1$st registers the computation of the compatible encoding $|zn\rangle|0\rangle\mapsto|zn\rangle|\mathsf{enc}_N(zN)\rangle$ for every $n\in N$:
$$\boldsymbol{|\phi_0\rangle^{\otimes s(t+1)}|\phi_1\rangle^{\otimes s(t+1)}|uH\rangle}|N\rangle^{\otimes 2t}|0\rangle.$$

[3] We explicitly mention here the encoding used for the last register in order to avoid any ambiguity in the notation. Observe also that z has the same encoding as zN ($\mathsf{enc}_N(zN)=\mathsf{enc}(z)$).

Theorem 4.11. *Let G be a black-box solvable group with unique encoding of length ℓ and let N be a normal subgroup of G such that G/N is abelian. Let α be a group action of G and let s, t be positive integers. Then* $\mathtt{TCS}(G, \alpha, s(t+1))$ *implements* Translating Coset Superposition$(G, \alpha, s(t+1))$ *using* $(4t+1)$ *implementations of* Translating Coset Superposition(N, α, s)*,* 2 *implementations of* Translating Coset Superposition$(G/N, (\alpha^s)_N, t)$*,* $2t$ *copies of* $|N\rangle$ *as ancilla, and quantum time* $\operatorname{poly}(\ell ts)$ *for the remaining computation.*

Proof. The complexity analysis of the algorithm is direct; only its analysis needs to be detailed. First observe that when the translating coset of $|\phi_0\rangle$ and $|\phi_1\rangle$ is empty, the algorithm sets $H = \emptyset$, and its correctness is clear.

From now on, we assume that the translating coset is not empty, and it is uH, for some unknown $u \in G$, where H is the unknown stabilizer of $|\phi_0\rangle$ and $|\phi_1\rangle$. Note that the translating coset of $|N \cdot \phi_0\rangle$ and $|N \cdot \phi_1\rangle$ for α_N in G/N is uHN/N, and therefore nonempty. At step 1, Theorem 4.10 is applied, then after step 4, the algorithm has therefore computed the state $|uHN/N\rangle$ in its last register.

In step 5, the group element in the last register is encoded using enc_N. But when its inverse is applied to the first register as an element of G, we mean to use the encoding enc. Thanks to our definition of compatible encoding, this makes sense as long as z satisfies $\mathsf{enc}_N(zN) = \mathsf{enc}(z)$. That is why the computed state becomes a uniform superposition of states $|z^{-1} \cdot \phi_0\rangle^{\otimes s} \dots |\phi_1\rangle^{\otimes s} \dots |\mathsf{enc}_N(zN)\rangle$, where the superposition is over $zN \in uHN/N$, and z is chosen such that $\mathsf{enc}_N(zN) = \mathsf{enc}(z)$. For each such z, we prove that states $|z^{-1} \cdot \phi_0\rangle$ and $|\phi_1\rangle$ have the translating coset $n_z(H \cap N)$ over the subgroup N for some $n_z \in N$ such that $|n_z \cdot \phi_1\rangle = |z^{-1} \cdot \phi_0\rangle$, meaning that $zn_z \in uH$.

Indeed, since $|u \cdot \phi_1\rangle = |\phi_0\rangle$, we get $|(z^{-1}u) \cdot \phi_1\rangle = |z^{-1} \cdot \phi_0\rangle$. Therefore $|z^{-1} \cdot \phi_0\rangle$ and $|\phi_1\rangle$ have the translating coset $z^{-1}uH$ over G. Since $zN \in uHN/N$, one can write $zn_z = uh_z$ for some $h_z \in H$ and $n_z \in N$. Note that both h_z and n_z are uniquely defined up to some element in $H \cap N$. Then the translating coset can be rewritten as n_zH, implying that $|z^{-1} \cdot \phi_0\rangle$ and $|\phi_1\rangle$ have a nonempty translating coset over N, which is $n_z(H \cap N)$.

Now set $H_1 = \bigcup_z \big(h_z(H \cap N)\big)$. Then after step 9, the state of the input register is $|uH_1\rangle$. The end of the proof consists in proving that $H_1 = H$. First observe that by definition $H_1 \subseteq H$. For the reverse inclusion, define for every $h \in H$ the coset $zN = uhN \in uHN/N$. Choose a representative z of zN such that $\mathsf{enc}_N(zN) = \mathsf{enc}(z)$. Since by construction $zN = uhN = uh_zN$, we get $h_z(H \cap N) = h(H \cap N)$, and therefore $h \in H_1$. □

If $|\phi_1\rangle = |\phi_0\rangle$, then $|N \cdot \varphi_1\rangle = |N \cdot \varphi_0\rangle$ as well. Therefore the same proof shows the following.

Theorem 4.12. *Let G, N, α, s, and t be as in Theorem* 4.11. *Then* $\mathtt{TCS}(G, \alpha, s(t+1))$ *implements* Stabilizer Superposition$(G, \alpha, s(t+1))$ *using as black boxes* $(4t+1)$ *implementations of* Translating Coset Superposition (N, α, s)*,* 2 *implementations of* Stabilizer Superposition$(G/N, (\alpha^s)_N, t)$*,* $2t$ *copies of* $|N\rangle$ *as ancilla, and quantum time* $\operatorname{poly}(\ell ts)$ *for the remaining computation.*

4.4. Applications to various groups. In this section, we study the consequences of the self-reducibility of Translating Coset for various families of solvable groups. We start by proving the following technical statement.

Theorem 4.13. *Let G be a solvable black-box group with unique encoding of length ℓ and let α be a group action of G on Γ. Assume that we are given a subnormal series $G = G_0 \rhd G_1 \rhd \cdots \rhd G_{r-1} \rhd G_r = \{1_G\}$ such that for* *every $1 \leq i \leq r$, either the factor group is elementary abelian of prime exponent bounded by e, or G_{i-1}/G_i is an abelian group of order at most s. Let $T = \big(\big(\log|G| + e + 2^{\sqrt{\log s}}\big)^{\Omega(e)} \log(1/\varepsilon)\big)^r$. Then there exists a quantum circuit that implements* Translating Coset Superposition(G, α, T) *with error ε in quantum time* poly(ℓT).

Proof. We actually show that, given a subnormal series $G = G_0 \rhd G_1 \rhd \cdots \rhd G_{r-1} \rhd G_r = \{1_G\}$ such that for every $1 \leq i \leq r$, the factor group G_{i-1}/G_i is either isomorphic to $\mathbb{Z}_{p_i}^{n_i}$, where p_i is a prime not greater than e and $n_i < n$, or G_{i-1}/G_i is an abelian group of order at most s; then, for $T = \big(\big(r+n+e+2^{\sqrt{\log s}}\big)^{\Omega(e)} \log(1/\varepsilon)\big)^r$, there exists a quantum circuit that implements Translating Coset Superposition(G, α, T) with error ε in quantum time poly(ℓT). From this the assertion follows as n_i and r are obviously bounded by $\log|G|$.

Set $u = \big(n + e + 2^{\sqrt{\log s}}\big)^{\theta(e)}$ so that for every prime $p \leq e$ and integer $0 < n' \leq n$, for every $\varepsilon > 0$ and for every permutation action α', by Corollary 4.5 `ElementaryAbelianTCS`$(\mathbb{Z}_p^{n'}, \alpha', \lfloor u \log \frac{1}{\varepsilon} \rfloor - 1)$ implements Translating Coset Superposition$(\mathbb{Z}_p^{n'}, \alpha', \lfloor u \log \frac{1}{\varepsilon} \rfloor - 1)$ with error less than $\varepsilon/4$ and also, for every abelian group A of size at most s, Translating Coset Superposition$(A, \alpha', \lfloor u \log \frac{1}{\varepsilon} \rfloor - 1)$ is implemented by `ArbitraryAbelianTCS`$(A, \alpha', \lfloor u \log \frac{1}{\varepsilon} \rfloor - 1)$ (by Corollary 4.6) with error at most $\varepsilon/4$ in quantum time less than $c_1(u \log \frac{1}{\varepsilon})^{d_1}$.

Define $\varepsilon_1 = \varepsilon$ and $\varepsilon_{j+1} = \varepsilon_j/(9u \log \frac{1}{\varepsilon_j})$. Put $T' = \prod_{j=1}^{r} \lfloor u \log \frac{1}{\varepsilon_j} \rfloor$. We define a circuit by induction on r that implements Translating Coset Superposition(G, α, T') with error at most ε. In the base case $r = 1$ we use either of the two circuits discussed above.

For $r > 1$, we construct the circuit by induction. Put $s = \prod_{j=2}^{r} \lfloor u \log \frac{1}{\varepsilon_j} \rfloor$ and $t = \lfloor u \log \frac{1}{\varepsilon} \rfloor - 1$. Let $N = G_1$. Then, by Theorem 4.11, `TCS`$(G, \alpha, s(t+1))$ implements Translating Coset$(G, \alpha, s(t+1))$ using $(4t+1)$ implementations of Translating Coset(N, α, s), 2 implementations of Translating Coset$(G/N, (\alpha^s)_N, t)$, $2t$ copies of $|N_{r-1}\rangle$, and quantum time less than $c_2(\ell st)^{d_2}$.

By the assumption on u, Translating Coset$(G/N, (\alpha^s)_N, t)$ can be implemented with error less than $\varepsilon/4$ in quantum time less than $c_1(\ell ts)^{d_1}$. (The oracle for α^s is implemented by s applications of the oracle for α.) Now, by induction Translating Coset(N, α, s) can be implemented with error $\varepsilon/9t < \varepsilon/(8t+2)$ in quantum time $c(\ell s)^d$, using $\mathrm{O}(s)$ copies of $|N_i\rangle$ for $1 \leq i \leq r-2$. The overall error is clearly less than ε.

We show that $T' = \big((ru)^{O(1)} \log \frac{1}{\varepsilon}\big)^r$. To see this, observe that $\log \frac{1}{\varepsilon_{j+1}} = \log \frac{1}{\varepsilon_j} + \log 9u + \log\log \frac{1}{\varepsilon_j}$. By induction on j, we can show that $\log \frac{1}{\varepsilon_j} \leq j^2 u \log \frac{1}{\varepsilon}$ if u is larger than an appropriate constant. (Indeed, the induction hypothesis gives $\log \frac{1}{\varepsilon_{j+1}} \leq j^2 u \log \frac{1}{\varepsilon} + \log 9u + 2\log j + \log\log \frac{1}{\varepsilon} \leq (j+1)^2 u \log \frac{1}{\varepsilon}$ if u is sufficiently large.) Therefore $T' \leq u^r \prod_{j=1}^{r} \log \frac{1}{\varepsilon_j} \leq (ru)^{2r} \big(\frac{1}{\varepsilon}\big)^r$.

The quantum time is bounded by $c_2(\ell st)^{d_2} + c_1(\ell st)^{d_1} + c(4t+2)s^d < c\ell {T'}^d$ if c and d are sufficiently large. □

The theorem above gives a polynomial time algorithm for Translating Coset in abelian groups of constant exponent. More generally, we have the following.

Theorem 4.14. *Let G be an abelian black-box group with unique encoding of length ℓ and let α be a group action of G on Γ. Assume that G has a subgroup N of exponent at most e such that G/N has size an most s. Let $T = \big((\log|G| + e + 2^{\sqrt{\log s}})^{\Omega(e)} \log(1/\varepsilon)\big)^{\log e}$. Then there exists a quantum circuit that* *implements* Translating Coset Superposition(G,α,T) *with error* ε *in quantum time* poly(ℓT).

Proof. Using, for instance, [34], a decomposition of G as a direct product of cyclic subgroups $H_1,\ldots,H_m$ of prime power order $p_i^{\alpha_i}$ can be computed in quantum time poly(ℓ). Considering only indices i such that $p_i \le e$, by an exhaustive search we can find in time polynomial in $\log|G|^{O(e)}$ integers $\beta_i \le \alpha_i$ $(i=1,\ldots,m)$ subject to the constraint $\mathrm{lcm}\{p_i^{\beta_i} | i=1,\ldots,m\} \le e$ such that $\prod_{i=1}^m p_i^{\beta_i}$ is maximal. Then the sum G_1 of the subgroups $H_i^{p_i^{\alpha_i-\beta_i}}$ is the largest (by cardinality) subgroup of exponent at most e, and consequently $|G/G_1| \le s$. Let $e = q_2 \cdots q_r$, where q_i are not necessarily distinct primes, and let $G_{i+1} = G_i^{q_i}$ $(i=2,\ldots,r)$. Then $r \le \log e$ and we can apply Theorem 4.13 to the sequence $G > G_1 > \cdots > G_r$. □

Using a similar proof we obtain the following generalization.

Theorem 4.15. *Let* G *be a solvable black-box group with unique encoding of length* ℓ *and let* α *be a group action of* G *on* Γ. *Assume that* G *has derived length* m *and that for every index* $0 < i \le m$, *the factor of the subsequent derived subgroups* $\widetilde{G_{i-1}} = G^{(i-1)}/G^{(i)}$ *have a subgroup* $\widetilde{N_{i-1}}$ *of exponent at most* e *such that* $|\widetilde{N_{i-1}}/\widetilde{N_{i-1}}| \le s$. *Let* $T = \left(s\left((\log|G|+e)^{\Omega(e)}\log(1/\varepsilon)\right)^{\log e}\right)^m$. *Then there exists a quantum circuit that implements* Translating Coset Superposition(G,α,T) *with error* ε *in quantum time* poly(ℓT).

The following theorem describes the class of groups for which our methods give polynomial time hidden subgroup algorithms. Recall that a smoothly solvable group has constant derived length, and the factors $\widetilde{G_{i-1}} = G^{(i-1)}/G^{(i)}$ satisfy the condition of the preceding corollary with constant e and $s = \mathrm{poly}(|\log|G)$.

Theorem 4.16. Translating Coset *and* Hidden Translation *can be solved over smoothly solvable groups in quantum polynomial time. Furthermore,* Stabilizer *and* Hidden Subgroup *can be solved over solvable groups having a smoothly solvable commutator subgroup in quantum polynomial time.*

Proof. The first statement follows directly from the preceding theorem, using Proposition 2.2. For the second part we additionally use Theorem 4.12. □

By [17], every solvable group has derived series of length $m = \mathrm{O}(\log\log|G|)$. Using this result and Theorem 4.15, we get a quasi-polynomial quantum algorithm for all solvable groups of constant exponent.

Theorem 4.17. *Let* G *be a solvable black-box group with unique encoding of length* ℓ *and of constant exponent. Then* Hidden Translation(G) *can be solved in quantum time* $\ell^{\mathrm{O}(1)}(\log|G|)^{\mathrm{O}(\log\log|G|)}$. *Furthermore, the* Hidden Subgroup *can be solved in quantum time* $\ell^{\mathrm{O}(1)}(\log|G|)^{\mathrm{O}(\log\log|G|)}$ *in groups* G *for which* G' *has constant exponent.*

Finally, an application of Theorem 4.13 with $e = 1$ and $s = |G|$ to the derived series of a solvable group gives the following.

Theorem 4.18. *Let* G *be a solvable black-box group with unique encoding of length* ℓ. *Then* Hidden Subgroup(G) *and* Hidden Translation(G) *can be solved with constant error in quantum time* $\ell^{\mathrm{O}(1)}(\log|G|)^{\mathrm{O}(\sqrt{\log|G|\cdot\log\log|G|})}$.

Acknowledgments. We wish to thank Mark Ettinger and Peter Høyer for sharing their knowledge and ideas about Hidden Translation with us, and Martin Rötteler for several useful discussions on Hidden Subgroup. We are also grateful to Cris Moore, Csaba Schneider, and Yong Zhang for helpful discussions regarding a previous version of this article, and for pointing out reference [17]. Part of the work of the last three authors was done while visiting MSRI, Berkeley, and part of the work of the second author was done during visits at the CQT in Singapore.

REFERENCES

[1] D. Aharonov, *Quantum computation: A review*, in Annual Review of Computational Physics, Vol. VI, World Scientific, Singapore, 1998, pp. 1–77.

[2] D. Aharonov, A. Kitaev, and N. Nisan, *Quantum circuits with mixed states*, in Proceedings of the 30th ACM Symposium on Theory of Computing, 1998, pp. 20–30.

[3] D. Aharonov and A. Ta-Shma, *Adiabatic quantum state generation and statistical zero knowledge*, in Proceedings of the 35th ACM Symposium on Theory of Computing, 2003, pp. 20–29.

[4] L. Babai, G. Cooperman, L. Finkelstein, E. Luks, and A. Seress, *Fast Monte Carlo algorithms for permutation groups*, J. Comput. System Sci., 50 (1995), pp. 296–308.

[5] L. Babai and E. Szemerédi, *On the complexity of matrix group problems* I, in Proceedings of the 25th Annual IEEE Symposium on Foundations of Computer Science, 1984, pp. 229–240.

[6] D. Bacon, A. Childs, and W. van Dam, *From optimal measurement to efficient quantum algorithms for the hidden subgroup problem over semidirect product groups*, in Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science, 2005, pp. 469–478.

[7] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf, *Quantum fingerprinting*, Phys. Rev. Lett., 87 (2001), 167902.

[8] R. Beals, *Quantum computation of Fourier transforms over symmetric groups*, in Proceedings of the 29th ACM Symposium on Theory of Computing, 1997, pp. 48–53.

[9] K. Cheung and M. Mosca, *Decomposing finite abelian groups*, Quantum Inf. Comput., 1 (2001), pp. 26–32.

[10] D. P. Chi, J. S. Kim, and S. Lee, *Quantum algorithms for the hidden subgroup problem on some semi-direct product groups by reduction to abelian cases*, Phys. Lett. A, 359 (2006), pp. 114–116.

[11] A. M. Childs and W. van Dam, *Quantum algorithms for algebraic problems*, Rev. Modern Phys., 82 (2010), pp. 1–52.

[12] A. Denney, C. Moore, and A. Russell, *Finding conjugate stabilizer subgroups in* $PSL(2;q)$ *and related problems*, Quantum Inf. Comput., 10 (2010), pp. 282–291.

[13] M. Ettinger and P. Høyer, *On quantum algorithms for noncommutative hidden subgroups*, Adv. in Appl. Math., 25 (2000), pp. 239–251.

[14] M. Ettinger, P. Høyer, and E. Knill, *The quantum query complexity of the hidden subgroup problem is polynomial*, Inform. Process. Lett., 91 (2004), pp. 43–48.

[15] K. Friedl, G. Ivanyos, F. Magniez, M. Santha, and P. Sen, *Hidden Translation and Orbit Coset in quantum computing*, in Proceedings of the 35th ACM Symposium on Theory of Computing, 2003, pp. 1–9.

[16] D. Gavinsky, *Quantum solution to the hidden subgroup problem for poly-near-Hamiltonian groups*, Quantum Inf. Comput., 4 (2004), pp. 229–235.

[17] S. P. Glasby, *The composition and derived lengths of a soluble group*, J. Algebra, 120 (1989), pp. 406–413.

[18] D. Gottesman and I. Chuang, *Quantum Digital Signatures*, Technical report 0105032, Quantum Physics e-Print archive, 2001.

[19] M. Grigni, L. Schulman, M. Vazirani, and U. Vazirani, *Quantum mechanical algorithms for the nonabelian Hidden Subgroup Problem*, in Proceedings of the 33rd ACM Symposium on Theory of Computing, 2001, pp. 68–74.

[20] S. Hallgren, *Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem*, J. ACM, 54 (2007), 1206039; preliminary version in Proceedings of the 34th ACM Symposium on Theory of Computing, 2002, pp. 643–658.

[21] S. Hallgren, *Fast quantum algorithms for computing the unit group and class group of a number field*, in Proceedings of the 37th ACM Symposium on Theory of Computing, 2005, pp. 468–474.

[22] S. Hallgren, A. Russell, and A. Ta-Shma, *Normal subgroup reconstruction and quantum computation using group representations*, in Proceedings of the 32nd ACM Symposium on Theory of Computing, 2000, pp. 627–635.

[23] P. Høyer, *Efficient Quantum Transforms*, Technical report 9702028, Quantum Physics e-Print archive, 1997.

[24] Y. Inui and F. Le Gall, *An Efficient Algorithm for the Hidden Subgroup Problem over a Class of Semi-direct Product Groups*, Technical report 0412033, Quantum Physics e-Print archive, 2004.

[25] G. Ivanyos, *On solving systems of random linear disequations*, Quantum Inf. Comput., 8 (2008), pp. 579–594.

[26] G. Ivanyos, F. Magniez, and M. Santha, *Efficient quantum algorithms for some instances of the non-abelian hidden subgroup problem*, Internat. J. Found. Comput. Sci., 14 (2003), pp. 723–739.

[27] G. Ivanyos, L. Sanselme, and M. Santha, *An efficient quantum algorithm for the hidden subgroup problem in extraspecial groups*, in Proceedings of the 24th Annual Symposium on Theoretical Aspects of Computer Science 2007, Lecture Notes in Comput. Sci. 4393, 2007, pp. 586–597.

[28] G. Ivanyos, L. Sanselme, and M. Santha, *An efficient quantum algorithm for the hidden subgroup problem in nil-*2 *groups*, in Proceedings of the 8th Latin American Symposium (LATIN 2008), Lecture Notes in Comput. Sci. 4957, 2008, pp. 759–771.

[29] A. Kitaev, *Quantum Measurements and the Abelian Stabilizer Problem*, Technical report, Quantum Physics e-Print archive, 1995; available online from http://xxx.lanl.gov/abs/quant-ph/9511026.

[30] A. Kitaev, A. Shen, and M. Vyalyi, *Classical and Quantum Computation*, Grad. Stud. Math. 47, AMS, Providence, RI, 2002.

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

[32] C. Moore, D. Rockmore, and A. Russell, *Generic quantum Fourier transforms*, in Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms, 2004, pp. 771–780.

[33] C. Moore, D. Rockmore, A. Russell, and L. J. Schulman, *The power of basis selection in Fourier sampling: Hidden subgroup problems in affine groups*, in Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms, 2004, pp. 1106–1115; journal version available at http://xxx.lanl.gov/abs/quant-ph/0503095.

[34] M. Mosca and C. Zalka, *Exact quantum Fourier transforms and discrete logarithm algorithms*, Int. J. Quantum Inform., 2 (2004), pp. 91–100.

[35] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, UK, 2000.

[36] J. Preskill, *Quantum Information and Computation*, http://www.theory.caltech.edu/people/preskill/ph229 (1998).

[37] M. Püschel, M. Rötteler, and T. Beth, *Fast quantum Fourier transforms for a class of non-Abelian groups*, in Proceedings of the 13th AAECC Symposium on Applied Algebra, Algebraic Algorithms, and Error Correcting Codes, Lecture Notes in Comput. Sci. 1719, 1999, pp. 148–159.

[38] O. Regev, *Quantum computation and lattice problems*, SIAM J. Comput., 33 (2004), pp. 738–760.

[39] M. Rötteler and T. Beth, *Polynomial-Time Solution to the Hidden Subgroup Problem for a Class of Non-abelian Groups*, Technical report, Quantum Physics e-Print archive, 1998; available online from http://xxx.lanl.gov/abs/quant-ph/9812070.

[40] A. Schmidt and U. Vollmer, *Polynomial time quantum algorithm for the computation of the unit group of a number field*, in Proceedings of the 37th ACM Symposium on Theory of Computing, 2005, pp. 475–480.

[41] P. W. Shor, *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*, SIAM J. Comput., 26 (1997), pp. 1484–1509.

[42] D. R. Simon, *On the power of quantum computation*, SIAM J. Comput., 26 (1997), pp. 1474–1483.

[43] C. Sims, *Computation in Finitely Presented Groups*, Cambridge University Press, Cambridge, UK, 1994.

[44] W. van Dam, S. Hallgren, and L. Ip, *Quantum algorithms for some hidden shift problems*, in Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, 2003, pp. 489–498.

[45] J. Watrous, *Succinct quantum proofs for properties of finite groups*, in Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science, 2001, pp. 60–67.

[46] J. Watrous, *Quantum algorithms for solvable groups*, in Proceedings of the 33rd ACM Symposium on Theory of Computing, 2001, pp. 60–67.