Classical simulations of non-abelian quantum Fourier transforms

Diploma Thesis
Juan Bermejo Vega

December 7, 2011
Garching

First reviewer: Prof. Dr. J. Ignacio Cirac
Second reviewer: Prof. Dr. Alejandro Ibarra
Introduction

Since it was discovered in 1994 that quantum computers can efficiently solve problems consider to be intractable classically [1] such as factorising composite numbers and calculating discrete logarithms [2] new quantum algorithms with rather different applications and several examples of the celebrated and intriguing exponential quantum speed-ups have been found [3]. However, understanding the fundamental questions “what is the power of quantum computers compared to classical ones?” and “which quantum circuits are useful to achieve exponential speed-ups?” is still a major challenge of the fields of quantum computation and complexity theory.

Classical simulation of quantum computations [4, 5, 6, 7, 8] is a fruitful approach to compare the power of quantum and classical computers where the goal is to identify classes of quantum calculations that can be efficiently simulated with classical algorithms and, hence, can not offer exponential-savings of computational resources to solve mathematical problems. Investigating such families of quantum processes can improve our comprehension of the essential features of quantum mechanics that are responsible for quantum speed-ups or indicate where to look for new algorithmic primitives.

In this work we study the classical simulation of quantum Fourier transforms (QFTs), key operations used in the design of several quantum algorithms that are related to exponential-quantum speed-ups. These transforms have a group-theoretical interpretation that arises from the representation theory of finite groups. Their relevance in quantum computation comes from the fact that they can be realised as unitary gates and, therefore, implemented as quantum circuits.

Quantum Fourier transforms have several applications in quantum information processing. For instance, they can be used to create quantum superpositions and to design measurement schemes that exploit quantum interference effects [9]. The simplest examples of quantum Fourier transforms are those defined over abelian groups. A well-known abelian-QFT used in early quantum algorithms is the Hadamard gate, an important ingredient of Deutsch-Jozsa’s [9] and Simon’s [1] algorithms. Also, the abelian quantum Fourier transform of the group \( \mathbb{Z}_{2^n} \) defined as

\[
F_{2^n} := \frac{1}{\sqrt{2^n}} \sum_{a,b=0}^{2^n-1} \omega_{2^n}^{ab} |a\rangle\langle b|
\]

with \( \omega_{2^n} := \exp(2\pi/2^n) \) is a crucial operation used in phase estimation [10] and Shor’s algorithm for factoring and computing discrete logarithms [2, 10].

The classical simulation of abelian quantum Fourier transforms has already been studied. Preceding works [11, 12, 13] have shown that the quantum Fourier transform \( F_{2^n} \) can be efficiently simulated classically when its action on initial product-states is followed by a projective measurement. Also, the action of \( F_{2^n} \) on more general classes of initial states can also be simulated efficiently on a classical computer for certain suitable families of inputs: in [11, 12] quantum-states generated by log-depth circuits are considered; some classes of matrix-product-states and graph states are introduced in [13].

In parallel to these results, in this work we investigate classical simulations of non-abelian quantum Fourier transforms and find that there are families of non-commutative groups whose QFTs can, as well, be efficiently simulated classically. We restrict to a class of non-abelian groups related to the hidden subgroup problem, an important topic of research in Quantum Computation. The family we consider are the semi-direct products of abelian groups \( \mathbb{Z}_p \rtimes \varphi A \) where \( A \) is an abelian group and \( \varphi \) an automorphism of \( A \) of prime-order \( p \). These groups are connected to certain lattice-problems of interest [14] and have been an object of study in several investigations during the last decade [15, 16, 17, 18, 19, 20].
Our work has two main parts. First, we develop new efficient quantum circuits to compute quantum Fourier transforms over semi-direct groups of the family $\mathbb{Z}_p \rtimes \phi A$. Then, we study the classical simulation of these circuits. In our simulation, the initial quantum states can be arbitrary coset-states $|xH\rangle$ of the semi-direct groups $\mathbb{Z}_p \rtimes \phi A$. Coset-states are uniform superpositions over computational-basis states labelled by the elements of a subgroup $H$ of the semi-direct product $\mathbb{Z}_p \rtimes \phi A$

$$|xH\rangle := \frac{1}{\sqrt{|H|}} \sum_{h \in H} |xh\rangle$$

(2)

From the formula follows that coset-states can be large or small quantum superpositions. For instance, when $H$ is chosen to be the whole group the state is a maximal uniform superposition over the computational-basis and when the subgroup is trivial $H = \{0\}$ the state is a computational-basis state. In the scheme we consider these states are transformed using a non-abelian quantum Fourier transforms and the final state at the end of the computation can be measured using two different schemes known in the non-abelian HSP literature as Weak and Strong Fourier Sampling. In this context we find that for a family of semi-direct groups of the form $\mathbb{Z}_p \rtimes \phi A$, the action of a non-abelian quantum Fourier transforms on arbitrary coset-states $|xH\rangle$ followed by a weak or strong Fourier sampling measurement can be efficiently simulated classically. Our main results apply to several groups considered in the field of quantum algorithms [16, 17, 18, 19, 20].

As in previous results [11, 12, 13] our simulation scheme does not apply to all gates involved in Shor’s factoring algorithm or in other quantum algorithms solving instances of the hidden subgroup problem. Therefore, our results do not imply that quantum algorithms showing exponential quantum speed-ups can be efficiently simulated in a quantum computer (for a discussion confer [11, 12, 13]). Yet our observations show that non-abelian quantum Fourier transforms, which are relatively complex unitary gates, can be efficiently simulated in a classical computer. Moreover, our simulation methods work not only for initial computational-basis states but also for arbitrary coset-states of these groups which can be arbitrarily large quantum superpositions.

Structure of this thesis

The structure of this work is the following.

Part I: Fourier transforms in Quantum Computation

The first part of the thesis gives an introduction to the applications of quantum Fourier transforms (QFTs) in Quantum Computation and classical simulations.

- In chapter 1 we give an introduction to the main applications of Quantum Fourier Transforms in quantum algorithms, the rigorous definition of these transforms and some mathematical preliminaries that we will use throughout this work.

- In chapter 2 we survey the hidden subgroup problem (HSP), a pure mathematical problem defined over finite groups but also one of most-studied topics of research in quantum computation. We review first the HSP for abelian groups, a problem which can be solved efficiently on a quantum computer (making crucial use of the abelian Quantum Fourier transform) but believed to be intractable classically. In addition, this problem subsumes several historical quantum algorithms as special cases, among them Simon’s [1] and Shor’s [2] (the first known examples of exponential quantum speed-ups). The non-abelian HSP has also interesting applications in quantum algorithms, yet a quantum algorithm to solve this problem in full generality is not known. In this chapter we explain the relationship of semi-direct groups of the form $\mathbb{Z}_p \rtimes \phi A$ and the non-abelian HSP and the role that non-abelian quantum Fourier transforms play in quantum algorithms to solve the HSP over these groups.

1These definitions are reviewed in chapter 2
• In chapter 3 we give an introduction to the field of classical simulation of quantum computations: motivation, well-known results, applications; and some examples of interesting classes of quantum circuits that can be efficiently simulated classically. We comment previous results about classical simulation of abelian quantum Fourier transforms related to the present work.

Part II: Group Theory

The second part of the thesis we give an introduction to semi-direct products of the form $\mathbb{Z}_p \rtimes \varphi A$ and computational problems related to groups.

• In chapter 4 we define the semi-direct products of the form $\mathbb{Z}_p \rtimes \varphi A$ and survey the mathematical properties of these groups relevant to this work.

• In chapter 5 we present some classical algorithms to perform certain computational tasks over abelian groups and semi-direct product groups of the form $\mathbb{Z}_p \rtimes \varphi A$.

Part III: Simulation of semi-direct quantum Fourier transforms

The third part of the thesis is devoted to explain the main-results.

• In chapter 6 we show how to implement the quantum Fourier transforms of the semi-direct groups $\mathbb{Z}_p \rtimes \varphi A$ in a quantum computer. We explain in detail the main features of our quantum circuits and the representation-theoretical interpretation of their components. Our circuits are efficient for several non-abelian groups which have been studied in Quantum Computation [16, 17, 18, 19, 20].

• In chapter 7 we consider the classical simulation of the non-abelian quantum Fourier transforms introduced in chapter 6. We find that, for the family of groups of the form $\mathbb{Z}_p \rtimes \varphi A$ the action of a non-abelian QFT on an arbitrary coset-state of $\mathbb{Z}_p \rtimes \varphi A$ followed by measurement that can be Weak or Strong Fourier Sampling can be efficiently simulated classically.

• In chapter 8 we raise conclusions of our work and pose open-questions.

Part IV: Appendixes

This part includes technical content.

• In appendix A we survey some representation-theoretical tools we used to find the irreducible representations of the groups $\mathbb{Z}_p \rtimes \varphi A$.

• In appendix B we give the proofs of some mathematical properties of the non-abelian quantum Fourier transforms of the semi-direct products $\mathbb{Z}_p \rtimes \varphi A$ which are useful to implement Weak and Strong Fourier sampling.

• In appendix C we analyse the efficiency of some classical algorithms for groups we use throughout the work.

• In appendix D we explain how we derived the quantum circuits from chapter 6.
# Contents

## Introduction

1 Fourier Transforms in Quantum Computation

1 Fourier transforms as unitary gates

---

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>11</td>
</tr>
<tr>
<td>1.1.1</td>
<td>11</td>
</tr>
<tr>
<td>1.1.2</td>
<td>12</td>
</tr>
<tr>
<td>1.2</td>
<td>13</td>
</tr>
<tr>
<td>1.2.1</td>
<td>13</td>
</tr>
<tr>
<td>1.2.2</td>
<td>13</td>
</tr>
<tr>
<td>1.3</td>
<td>14</td>
</tr>
<tr>
<td>1.3.1</td>
<td>14</td>
</tr>
<tr>
<td>1.3.2</td>
<td>14</td>
</tr>
<tr>
<td>1.3.3</td>
<td>14</td>
</tr>
<tr>
<td>1.3.4</td>
<td>14</td>
</tr>
<tr>
<td>1.3.5</td>
<td>15</td>
</tr>
<tr>
<td>1.4</td>
<td>15</td>
</tr>
<tr>
<td>1.4.1</td>
<td>15</td>
</tr>
<tr>
<td>1.4.2</td>
<td>16</td>
</tr>
<tr>
<td>1.4.3</td>
<td>16</td>
</tr>
<tr>
<td>1.5</td>
<td>17</td>
</tr>
<tr>
<td>1.5.1</td>
<td>17</td>
</tr>
<tr>
<td>1.5.2</td>
<td>18</td>
</tr>
<tr>
<td>1.5.3</td>
<td>18</td>
</tr>
</tbody>
</table>

2 The Hidden Subgroup Problem

2.1 | 22 |
2.2 | 22 |
2.2.1 | 23 |
2.2.2 | 23 |
2.3 | 24 |
2.3.1 | 24 |
2.3.2 | 25 |
2.3.3 | 25 |
2.4 | 26 |

3 Classical simulation of quantum computers

3.1 | 27 |
3.2 | 28 |
3.3 | 28 |
3.3.1 | 28 |
# CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.2</td>
<td>Simulation of cyclic quantum Fourier transforms</td>
<td>30</td>
</tr>
<tr>
<td>3.4</td>
<td>Applications</td>
<td>30</td>
</tr>
<tr>
<td>II</td>
<td>Group Theory</td>
<td>33</td>
</tr>
<tr>
<td>4</td>
<td>Semi-direct products</td>
<td>35</td>
</tr>
<tr>
<td>4.1</td>
<td>Definition</td>
<td>35</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Opposite group</td>
<td>35</td>
</tr>
<tr>
<td>4.2</td>
<td>Arithmetic operations</td>
<td>36</td>
</tr>
<tr>
<td>4.3</td>
<td>Subgroups</td>
<td>36</td>
</tr>
<tr>
<td>4.4</td>
<td>Cosets</td>
<td>37</td>
</tr>
<tr>
<td>5</td>
<td>Computational complexity of group-theoretical problems</td>
<td>39</td>
</tr>
<tr>
<td>5.1</td>
<td>Conventions</td>
<td>39</td>
</tr>
<tr>
<td>5.2</td>
<td>Algorithms for abelian groups</td>
<td>40</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Computing abelian group automorphisms</td>
<td>40</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Advanced problems for abelian groups</td>
<td>41</td>
</tr>
<tr>
<td>5.3</td>
<td>Algorithms for semi-direct products</td>
<td>42</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Generating-sets</td>
<td>43</td>
</tr>
<tr>
<td>III</td>
<td>Simulation of semi-direct quantum Fourier transforms</td>
<td>45</td>
</tr>
<tr>
<td>6</td>
<td>Efficient QFTs for semi-direct product groups</td>
<td>47</td>
</tr>
<tr>
<td>6.1</td>
<td>Representation theory of semi-direct products</td>
<td>48</td>
</tr>
<tr>
<td>6.1.1</td>
<td>The $\phi$-partition of the abelian group $A$</td>
<td>48</td>
</tr>
<tr>
<td>6.1.2</td>
<td>The dual automorphism</td>
<td>48</td>
</tr>
<tr>
<td>6.1.3</td>
<td>Irreducible representations</td>
<td>49</td>
</tr>
<tr>
<td>6.1.4</td>
<td>Regular representations</td>
<td>49</td>
</tr>
<tr>
<td>6.2</td>
<td>An efficient canonical circuit for the semi-direct QFT</td>
<td>50</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Conventions</td>
<td>50</td>
</tr>
<tr>
<td>6.2.2</td>
<td>The quantum circuit $F$</td>
<td>51</td>
</tr>
<tr>
<td>6.2.3</td>
<td>The unitary gate $F$ implements a canonical quantum Fourier transform</td>
<td>52</td>
</tr>
<tr>
<td>6.2.4</td>
<td>Block-diagonal formulas</td>
<td>53</td>
</tr>
<tr>
<td>6.3</td>
<td>Weak and Strong Fourier Sampling</td>
<td>54</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Weak Fourier sampling</td>
<td>54</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Strong Fourier sampling</td>
<td>55</td>
</tr>
<tr>
<td>6.4</td>
<td>QFTs for arbitrary choices of basis</td>
<td>55</td>
</tr>
<tr>
<td>6.5</td>
<td>Conclusions and relation to previous work</td>
<td>56</td>
</tr>
<tr>
<td>7</td>
<td>Classical simulation of non-abelian QFTs</td>
<td>59</td>
</tr>
<tr>
<td>7.1</td>
<td>Our family of non-abelian groups</td>
<td>60</td>
</tr>
<tr>
<td>7.2</td>
<td>Quantum coset states</td>
<td>60</td>
</tr>
<tr>
<td>7.2.1</td>
<td>The hierarchy of semi-direct coset-states</td>
<td>60</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Simulating projective measurements on cosets-states</td>
<td>61</td>
</tr>
<tr>
<td>7.3</td>
<td>Preliminaries</td>
<td>62</td>
</tr>
<tr>
<td>7.3.1</td>
<td>Decomposition of the circuit $F$</td>
<td>62</td>
</tr>
<tr>
<td>7.3.2</td>
<td>Simulation of abelian cosets</td>
<td>63</td>
</tr>
<tr>
<td>7.3.3</td>
<td>An important lemma</td>
<td>63</td>
</tr>
<tr>
<td>7.4</td>
<td>Non-abelian Fourier transforms can be classically simulated</td>
<td>64</td>
</tr>
<tr>
<td>7.5</td>
<td>Simulating weak and strong Fourier sampling</td>
<td>66</td>
</tr>
<tr>
<td>7.5.1</td>
<td>General case</td>
<td>66</td>
</tr>
</tbody>
</table>
Part I

Fourier Transforms in Quantum Computation
Chapter 1

Fourier transforms as unitary gates

Summary

Quantum Fourier transforms are among the most important unitary gates in Quantum Computation and they are extensively used in quantum-algorithm-design. In this chapter we define these quantum operations, introduce their best-known applications and give some mathematical preliminaries that we will use throughout this thesis.

The structure of this chapter is the following. In section 1.1 we discuss in a non-technical way the role quantum Fourier transforms play in quantum computation. In section 1.2 we survey some basic notation and conventions used in the Circuit Model of Quantum Computation. In section 1.3 we give an introduction to representation theory of finite groups. In section 1.4 we define the quantum Fourier transform for any finite group and explain its main properties. Finally, in section 1.5 we explain the abelian quantum Fourier transform and some character theory.

1.1 The role of QFTs in quantum algorithms

The Fourier transform is a famous mathematical operation which acts over spaces of functions, well-known for its applications in physics, computer science, mathematics and engineering. The transform is commonly defined as a map that sends function a $f : \mathbb{R} \to \mathbb{C}$ into a new one $\hat{f} : \mathbb{R} \to \mathbb{C}$ in an invertible way summarised by the equations

$$
\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} f(x) e^{2\pi i \omega x} dx \\
f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \hat{f}(\omega)e^{-2\pi i \omega x} d\omega
$$

As the transform is invertible it defines a duality. This transforms are continuous but the Fourier transform can be defined also in discrete spaces such as finite groups using representation theory. These discrete group-theoretical Fourier transforms can be written as unitary matrices and, therefore, they can realised as quantum circuits in a quantum computer. It is natural to wonder whether quantum Fourier transforms could be have applications in quantum information processing.

In fact, it turns out that quantum Fourier transforms are among the most important unitary gates in Quantum Computation and a basic-tool in quantum-algorithm design. In particular, they are one of the crucial ingredients of several algorithms showing exponential quantum speed-ups such as Shor’s quantum algorithm for order-finding, which can be applied to factorise composite numbers and compute discrete logarithms [2], two problems that are believed to be intractable for classical computers.

Two applications of this transform are particularly important in quantum computation. The first one is phase estimation a useful quantum routine that estimates the eigenvalue of an eigenstate of an unitary gate and has several applications for rather different quantum-informational tasks (section 1.1.1). The second one is Fourier Sampling the measuring scheme used to solve the abelian hidden subgroup problem (section 1.1.2).
CHAPTER 1. FOURIER TRANSFORMS AS UNITARY GATES

1.1.1 Quantum phase estimation

Quantum phase estimation [10] is an algorithm which solves the following problem. One is given two inputs: the first, a unitary gate $U$ together with a black-box (a subroutine) which efficiently computes the controlled-$U^j$ gates where $j$ is an integer; the second, a quantum state $|u\rangle$ which is promised to be an eigenstate of $U$ with eigenvalue $e^{2\pi i \theta_u}$, where the phase value of the phase $\theta_u$ is not told to us. Then, the phase estimation algorithm finds an estimation of the phase of the eigenvalue, $\theta_u$.

We sketch the main ideas used in this algorithm. Assume for simplicity, that the binary representation of this number has exactly $n$-digits.

$$\theta_u = 0.\theta_1\theta_2\ldots\theta_n = \sum_{i=1}^{n} 2^{-i}\theta_i$$ (1.1)

Then the phase estimation algorithm uses the given black-box to create a state with the following form:

$$|\psi\rangle := \frac{1}{\sqrt{n}} \sum_{a=0}^{2^n-1} \omega_{2^n}^{a\theta_u} |a\rangle |u\rangle$$ (1.2)

where he have an $n$-qubit register, a register for the given eigenstate and $\omega_{2^n} := \exp (2\pi i / 2^n)$ is a primitive root of the unity. Then applying the inverse of the quantum Fourier transform $F_{2^n}^\dagger$ one obtains the value of the phase $\theta_u$ in the first register.

$$F_{2^n}^\dagger |\psi\rangle = |\theta_u\rangle |u\rangle$$ (1.3)

If the phase $\theta_u$ is an arbitrary real number one can use the above procedure to get an $n$-digits approximation of the value of the phase $\theta'$ with high probability. For a rigorous analysis [9] or [21] can be consulted.

Applications

In spite of its simplicity, phase estimation has turned out to be a very useful quantum algorithm. In particular, Shor’s algorithm can be view as an application of phase estimation (consult [21] for a review). Also, it is used as a subroutine in some recent algorithms to solve problems in quantum information and computation of rather different nature:

- A quantum algorithm [22] which estimates expected values of the form $x^T M x$ where $x$ is the solution of a linear system of equations $Ax = b$. For certain types of matrices this algorithm shows quantum exponential speed-ups.
- Quantum Metropolis sampling algorithm [23] which allows create the Gibbs state of an input Hamiltonian.
- A recent quantum algorithm [24] to prepare a particular class of injective Projected Entangled Pair States in a quantum computer.

1.1.2 Fourier sampling and the Hidden Subgroup Problem

It turns out that many known quantum algorithms based on quantum Fourier transforms such as Deutsch’s, Simon’s and Shor’s and several others [9, 21] share a common mathematical structure: they are all particular instances of a problem defined over finite groups known as the abelian hidden subgroup problem (HSP) [9, 21, 25]. This problem can be solved efficiently in a quantum computer using an algorithm based on Fourier sampling which we review in chapter 2 yet it is believed to be intractable classically.

The abelian HSP can be generalised further to non-abelian finite groups. Unlike the abelian counterpart, there is no-known quantum algorithm to solve this problem efficient, although it would be highly interesting to find one since it is known that certain mathematical problems over lattices and the graph
isomorphism problem reduce to the non-abelian HSP over the dihedral \([14]\) and the symmetric group \([26]\) respectively. Owing to these facts, considerable effort has been done to study the non-abelian HSP during the last decade. Negative results suggest that the symmetric HSP could be a hard problem even for quantum computers and that it can not be solved using the Fourier sampling approach that worked in the abelian case \([27, 28, 29]\).

There are more reasons to believe that there could be a quantum algorithm that solves this problem efficiently, since the dihedral group is a semi-direct product of two abelian groups \(Z_2 \ltimes Z_N\). Colloquially speaking semi-direct products of the form \(H \times A\) are generalisations of direct products \(H \times A\) using a non-commutative operation instead of the addition. Intuitively, the HSP over these groups looks simpler since their structure resembles abelian direct-products', making this class an interesting candidate to study the non-abelian HSP. Moreover, efficient quantum algorithm have been found for non-commutative groups inside this family and, in the case of the dihedral group, there is a quantum algorithm to solve the HSP though its inefficient \([16, 15]\). To this latter family belong the groups \(Z_p \ltimes \varphi A\), which are the object of study of this thesis.

\section{The circuit model}

We introduce now some basic notation and conventions related to the Circuit Model of Quantum Computation, that we will keep and use throughout the whole work.

\subsection{G-states and the Hilbert space \(\mathcal{H}_G\)}

Throughout this work many quantum-states will live on a Hilbert space \(\mathcal{H}_G\) whose computational-basis states \(|g\rangle\) are labelled by the elements of a finite group \(G\).

\[|\psi_G\rangle := \sum_{g \in G} c_g |g\rangle \quad (1.4)\]

Quantum states \(|\psi_G\rangle\) of this form are sometimes called “G-states” to remind this feature. As the number of computational-basis states \(|g\rangle\) matches the \textit{order} of the group \(|G|\) (i.e. the total number of elements \(g \in G\), our states can be written down using at most \(n := \lceil \log |G| \rceil\) qubits. The number \(n\) will be used as a measure of the size of our physical system and the amount of memory required to store our quantum states.

In this work logarithms are taken in basis 2. The symbols \([x]\) mean that the real number \(x\) is rounded to the closest, bigger, natural number (in absolute value).

\subsection{Quantum circuits and complexity}

The \textit{input-size} \(n\) will also be used throughout this work to measure the difficulty of the computational problems we will deal with. A classical operation over a \(n\)-bit register or a quantum operation over a \(n\)-qubit quantum state is said to be \textbf{efficient} if the time and memory resources needed to perform it scale \textit{at most} as a polynomial of \(n\). Throughout the work ‘efficient’ will be synonymous to ‘in polynomial time’ and we will use the notation \(\text{poly}(n)\) to say ‘a polynomial of \(n\).

We will also use the big \(O\) notation to upper-bound quantities. In this notation \(O(\text{poly}(n))\) means ‘asymptotic-growth scales at most as fast as a polynomial of \(n\).’ Equivalently, the symbol \(\Omega\) is used to lower-bound quantities. When upper- and lower-bounds are known to match asymptotically we will use the theta symbol \(\Theta\). A cheat-sheet with the main properties of the big \(O\) notation can be found in \([30]\).

In this work, when we refer to a quantum circuit, we will always implicitly mean a uniformly generated family of quantum circuits \([31]\). We say that a given quantum circuit acting on at most \(n\)-qubit is \textbf{efficient} if it contains \(O(\text{poly}(n))\) elementary quantum gates. By ‘elementary’ quantum gates we mean gates taken from an universal set of quantum gates universal for quantum computation, e.g. the Hadamard and the Toffoli gate \([32, 33]\). Consult \([9]\) or \([34]\) for more information on universal sets of quantum gates.
1.3 Representation theory preliminaries

To understand the quantum Fourier Transform over finite groups we need to review some representation theory. From now, whenever we talk about groups we will mean finite groups. For any finite group an representation can be chosen to be unitary (as well as their changes of basis) \[35\] and we will always make this assumption.

1.3.1 Representation

A linear representation (or simply representation) \( \sigma \) of a finite group \( G \) of dimension \( d \) is a homomorphism \( \sigma : G \rightarrow U(d) \) where \( U(d) \) is the group of \( d \times d \) unitary matrices with matrix-multiplication as operation. This means that for any element \( g \) in \( G \) there exists a \( d \times d \) unitary matrix \( \sigma(g) \) such that for any \( g, h \) from \( G \)

\[
\sigma(gh) = \sigma(g)\sigma(h)
\]  

(1.5)

1.3.2 Regular representations

If the representation is a one-to-one correspondence the group \( G \) is isomorphic to the set of matrices \( \sigma(g) \) and the representation is called faithful. Two important faithful representations are the so-called left and right regular representations. If the order of the group is \( N = |G| \), the regular representations are \( N \times N \) unitary matrices acting on the Hilbert space \( \mathcal{H}_G \) labelled by group elements of \( G \) and defined as follows:

\[
X_L(g)|h\rangle := |gh\rangle
\]

(1.6)

\[
X_R(g)|h\rangle := |hg^{-1}\rangle
\]

(1.7)

1.3.3 Equivalence and reducibility

Representations that are equal up to a change of basis are called equivalent (and inequivalent if not). If for a given representation \( \sigma \) the matrices \( \sigma(g) \) can be simultaneously block-diagonalised into new matrices \( \sigma_1(g) \oplus \sigma_2(g) \) (where the size of the blocks is fixed for all of them) the representation \( \sigma \) is said to be reducible. Then each of the new blocks \( \sigma_1, \sigma_2 \) are representations of smaller dimension.

Representations which are not reducible are called irreducible representations or irreps. Irreducible representations play a central role in Fourier Analysis of finite groups and we will study some of their properties. For each finite group \( G \) the complete set of all possible irreducible inequivalent representations \( \hat{G} \) is finite\(^1\).

1.3.4 Character functions

In many situations it is desirable to work with the algebraic properties of a representation \( \sigma \) without having to choose a particular basis to express the matrices \( \sigma(g) \). A useful tool in this context are the so-called character functions defined taking the trace of the irrep matrices returning a function independent of basis. The character function \( \chi_\sigma \) of the representation \( \sigma \) is a function which sends the elements of the group \( G \) into the complex numbers:

\[
\chi_\sigma(g) := \text{tr}[\sigma(g)]
\]

(1.8)

By definition all one-dimensional (hence, irreducible) representations are character functions.

\(^1\)The order of this set equals the number of conjugacy classes of \( G \).
1.3.5 A remark on characters and basis

In sometimes convenient in some branches of physics of mathematics to work exclusively with character functions and forget about the basis of the representations. This is not the case in Quantum Computation as we will usually embed the unitary gates defined by a representation inside a quantum circuit and a particular basis for them must always be chosen. Moreover, some particular basis might be more appropriate to design efficient quantum circuits, something that makes harder to design good circuits. We will come back to these details later, especially in chapter 6 where we talk about implementations of quantum Fourier transforms.

1.4 The Fourier transform of a finite group

1.4.1 Definition

The left and right regular representations of a finite group $G$ (1.6) are particular, not only because they define matrix groups isomorphic to $G$, but also because they ‘contain’ all irreducible representations of the group. Rigorously, one can prove [9, 35] that both regular representations are reducible for any non-trivial finite group $G$, and that any irreducible representation $\sigma$ of dimension $d_\sigma$ of $G$ must appear exactly $d_\sigma$ in the block diagonal decomposition of both left and right regular-representations. As a consequence one obtains the following equality:

$$|G| = \sum_{\sigma \in \hat{G}} d^2_\sigma (1.9)$$

Then, there exists a change of basis $U$ such that the new representations $Z_L/R(g) := UX_{L/R}(g)U^\dagger$ are $|G| \times |G|$ block-diagonal unitary gates with the structure of blocks we have described. A unitary change of basis with this properties is called a quantum Fourier transform of $G$.

**Definition 1.4.1.** For a given finite group $G$, a **quantum Fourier transform** of $G$ is any unitary gate $U$ performing the change of basis which decomposes both the left and right regular representations of the group into block-diagonal matrices with the following structure:

$$Z_L(g) := UX_L(g)U^\dagger = \bigoplus_{\sigma \in G} I_{d_\sigma} \otimes \sigma(g) \quad (1.10)$$

$$Z_R(g) := UX_R(g)U^\dagger = \bigoplus_{\sigma \in G} \sigma(g)^* \otimes I_{d_\sigma} \quad (1.11)$$

Where each irreducible representation $\sigma$ of dimension $d_\sigma$ is repeated exactly $d_\sigma$ times in the direct-sum decomposition.

It follows from the definition that there is not only one unique quantum Fourier transform for groups with high-dimensional irreducible representations, since the matrices of those can be equivalently represented in any arbitrary basis. As a consequence, the quantum Fourier transform of a finite group $G$ is **uniquely defined** if and only if the group $G$ is **abelian** (up to re-orderings of the labels of the irreducible representations). This is a consequence of Schur’s lemma which implies that all the irreps of an abelian group are one-dimensional [25, 35], i.e. character functions, and a “change of basis” of a function consists in multiplying and dividing it by the same complex number leaving it invariant. Moreover, any non-abelian group must have at least one high-dimensional irreducible representation or otherwise the regular-representation matrices $Z_{L/R}(g)$ would be diagonal and commute, contradicting that they form a group isomorphic to the original one.

**Convention.** In this work the super-index $*$ denotes complex-conjugate, like in $\sigma(g)^* = \sigma(g)^{T^*} = \sigma(g^{-1})^T$ or in $\chi_\sigma^*(g) = \chi_\sigma(g^{-1})$. For $p$th-roots of the unity we usually use the notation $\omega^T = \omega^{-1}$ instead.
CHAPTER 1. FOURIER TRANSFORMS AS UNITARY GATES

1.4.2 The QFT is a unitary gate

The reason why these transforms are called “called” quantum is that they are unitary gates \[^35\] and can, in principle, be implemented as quantum circuits in a quantum computer. In fact, there is a canonical way to construct a unitary gate that implements a non-abelian Fourier transform for any finite group \(G\), which we explain now. For each irreducible representation \(\sigma \in \hat{G}\) denote by \([\sigma(g)]_{i,j}\) the coefficient with row \(i\) and column \(j\) of the matrix \(\sigma(g)\). There are \(|G|\) of such coefficients because of equation 1.9 and they can be used to define a new computational-basis labelled by states \(|i,j,\sigma\rangle\). Then the following unitary gate is a non-abelian quantum Fourier transform of \(G\):

\[
F_G := \sum_{g \in G} \sum_{\sigma \in \hat{G}} \sqrt{\frac{d_{\sigma}}{|G|}} \sum_{i,j \in 1} [\sigma(g)]_{j,i} |i,j,\sigma\rangle \langle g|
\]  

It can be checked that this transform is a unitary transformation using some expressions known as ‘orthogonality rules of the irreducible representations’. Although we will not work explicitly with this equations throughout the thesis, we include them here for the sake of completeness:

\[
\sum_{g \in G} [\sigma_1(g)]_{ij} [\sigma_2(g)]_{kl}^* = \frac{|G|}{d_{\sigma_1}} \delta_{i,k} \delta_{j,l} \delta_{\sigma_1,\sigma_2}
\]  

These equations generalise the orthogonality rules we gave for the abelian characters (equation 1.16) and can be derived directly from Schur’s lemma \[^35\]. The reference \[^25\] contains a derivation showing how the matrix (6.18) block-diagonalises the regular representations as in definition 1.4.1, proving that it is indeed a Fourier transform. We include this derivation as well for the QFTs we study in this work in appendix B.

A final observation. It is clear from equation 6.18 that unitary gates non-abelian quantum Fourier transforms can be very different for groups with substantially different irreducible representations. For these reason, a normal approach is to study the properties of this transform for classes of groups which share some common features. As an example, finite abelian groups form a family of groups which decompose as finite direct-products of finite cyclic groups.

1.4.3 Implementation as quantum circuits

From previous section it is already clear that the definition 1.4.1 does not provide a constructive method to find efficient quantum circuits to implement quantum Fourier transforms over finite groups. However, this topic has been studied during the last decades and it is currently known how to efficiently implement quantum Fourier transforms over several families of finite groups.

First, for the abelian QFT, well-known efficient quantum circuits exist. Abelian quantum Fourier transforms are one of the most important unitary gates that have been used in quantum computation and it plays a key-role in quantum algorithms such as phase estimation and Shor’s Factoring algorithm \[^9\] \[^25\]. We dedicate the next section (1.5) to show how the abelian-QFT unitary gates look like, comment the main ideas used to implement them as quantum circuits, as well as some classical-simulation results concerning abelian quantum Fourier transforms which generalise the Gottesman-Knill theorem to abelian groups.

For non-abelian groups designing QFTs is a rather tougher subject, yet several techniques exist to approach this task. Efficient quantum circuits for the non-abelian QFT have been explicitly written down for several types of metacyclic and semi-direct product groups including the dihedral group \[^55\] \[^57\], the generalised quaternion group \[^57\], the Heisenberg group \[^56\], and some wreath-product groups \[^17\]. The
techniques used in these works were generalised further in [38] where a generic method to find efficient QFTs for even more exotic families is presented. Summarising all these techniques lies beyond the scope of this thesis and we will not discuss about them.

1.5 The abelian quantum Fourier transform

In this section we given an overview of how the abelian QFT unitary gate looks like and comment the key-ideas used to implement it efficiently as a quantum circuit.

1.5.1 Character functions of abelian groups

Now we review the fundamental properties of the representation theory of commutative groups, which is widely used in the field of quantum algorithms. From now on, any particular abelian group \( A \) will given to us in a canonical form, decomposed as a product of smaller cyclic groups:

\[
A = \mathbb{Z}_{N_1} \times \ldots \times \mathbb{Z}_{N_d} \tag{1.14}
\]

Here, \( \mathbb{Z}_N \) denotes the group of numbers from 0 to \( N - 1 \) with addition modulo \( N \). Physically, each cyclic factor \( \mathbb{Z}_{N_i} \) can be understood as a \( N_i \)-levels subsystem of a bigger system \( A \). This form of \( A \) is general for finite abelian groups since the fundamental classification theorem of finite groups states that any abelian group is isomorphic to a product of cyclic factors. However, there are not known efficient classical algorithms to compute this decompositions (in contrast with quantum computers, for which an efficient algorithm for this task exists [39]) and we will assume throughout this work that such decomposition is given to us as an input.

The complete set of inequivalent irreducible representations of the abelian group \( A \) is easy to describe from the decomposition (1.14) and using that irreducible representations of abelian groups are always one-dimensional homomorphic functions. Then any homomorphic function from the group \( A \) into the complex numbers must satisfy \( f(0, 1, \ldots, 0) = 1 \) for all values of \( i \), because \( (0, 1, \ldots, N_i, \ldots, 0) = 0 \) and the neutral element of the original space must be mapped to the neutral of the final space. Then a complete set of solutions of this system of equations is given by all character functions of the form:

\[
\chi_a(b) := \exp \left\{ 2\pi i \left( \frac{a_1 b_1}{N_1} + \ldots + \frac{a_d b_d}{N_d} \right) \right\} \tag{1.15}
\]

Where \( a \) is any element of \( A \). Moreover, \( \chi_a \) and \( \chi_{a'} \) are equivalent representations if and only if they are equal as functions which happens if and only if \( a = a' \).

Fundamental properties of the characters

We end this section with some properties of the character functions that will be frequently used later.

(a) Pontryagin duality. The set of all inequivalent irreducible representations \( \hat{A} \) of an abelian group \( A \) is a group with the operation \( \chi_{a_1} \cdot \chi_{a_2} = \chi_{a_1 + a_2} \). This group is isomorphic to \( A \) via the identification \( \chi_a \leftrightarrow a \) and it is usually called the dual group (or character group) of \( A \).

(b) Orthogonality rules of the characters.

\[
\sum_{b \in A} \chi_{a_1}(b) \chi_{a_2}^*(b) = \delta_{a_1, a_2} |A| \tag{1.16}
\]

(c) For any two elements \( a, b \) from \( A \) the following symmetry holds:

\[
\chi_a(b) = \chi_b(a) \tag{1.17}
\]

Commutative groups have some representation-theoretically benign properties that simplify their Fourier analysis, some of have already been commented in section [1.5.1]. In particular, for any abelian group \( A \):
• Left and right regular representations coincide \( X_A \equiv X_L \equiv X_R \). We will just talk about ‘the regular representation’ of this groups and denote it by \( X_A \).

• All irreducible representations are one-dimensional, the abelian Fourier Transform is uniquely defined and the Fourier-transformed \( Z \) regular-representation matrices (1.10, 1.11) are diagonal.

• The group \( A \) is isomorphic to the character group \( \hat{A} \). This guarantees that applying a quantum Fourier transform twice makes sense algebraically.

1.5.2 Abelian gates

The unitary-gate implementing the regular-representation of \( A \) is (cf. equation 1.18):

\[
X_A(a) = \sum_{b \in A} (a + b) |b \rangle \langle b |
\]

(1.18)

Using equation 6.18 the quantum Fourier transform of any abelian group can be written as:

\[
F_A := \frac{1}{\sqrt{|A|}} \sum_{a,b \in A} \chi_a(b) |a \rangle \langle b |
\]

(1.19)

It is normal to present this formula without referencing the dual group \( \hat{A} \) in the abelian case since this group is isomorphic to \( A \). Now we show that this gate effectively block-diagonalises the matrices \( X_A(a) \) as a direct-sum of the characters functions of the group, i.e., the one-dimensional representations of the group.

\[
Z_A(a) := F_A X_A(a) F_A^* = |A|^{-1} \sum_{c,c' \in A} \chi_c(a + b) \chi_c^*(b) |c \rangle \langle c' |
\]

(1.20)

\[
= |A|^{-1} \sum_{c,c' \in A} \chi_c(a) \left( \sum_{b \in A} \chi_c(b) \chi_c^*(b) |c \rangle \langle c' | \right) = \sum_{c,c' \in A} \chi_c(a) \delta_{c,c'} |c \rangle \langle c' |
\]

(1.21)

\[
= \sum_{c \in A} \chi_c(a) |c \rangle \langle c |
\]

(1.22)

Where we have used the orthogonality rule of the characters from eq. (1.16). Since the characters \( \chi_c \) form a complete set of inequivalent irreducible representations of \( A \), the Fourier transformed regular representation \( Z_A(a) \) has the structure given in definition 1.4.1 and it follows that the gate \( F_A \) performs the (unique) quantum Fourier transform of the abelian group \( A \).

The abelian quantum Fourier transform has yet another good property: if we apply the \( F_A \) change of basis twice over the regular representation \( X(a) \) we would get the inverse \( X(a) \) of the original matrix. In fact, the squared Fourier transform \( U_{(-)} := F_A^2 \) is a permutation gate which implements the sign-flip automorphism of \( A \), i.e. the gate that sends any element to its inverse:

\[
U_{(-)} := \sum_{a \in A} |a \rangle \langle -a |
\]

(1.23)

1.5.3 Tensor product formulas

As we commented at the beginning of the chapter, any abelian group \( A \) has a decomposition as a direct product of cyclic factors \( A := \mathbb{Z}_{N_1} \times \ldots \times \mathbb{Z}_{N_d} \). This decomposition induces a tensor-product decomposition of our Hilbert space \( \mathcal{H}_A = \mathcal{H}_{\mathbb{Z}_{N_1}} \otimes \ldots \otimes \mathcal{H}_{\mathbb{Z}_{N_d}} \) which extends to the regular-representation gates of the group and the Fourier transform. This formulas allow to implement the QFT over an
arbitrary abelian group, reducing the problem to implementing the QFT for cyclic groups $\mathbb{Z}_N$. For the regular representation gates $X_A(a)$ one obtains the following expressions:

$$X_A(a) = \sum_{b \in A} |a + b\rangle \langle b|$$  \hspace{1cm} (1.25)

$$= \sum_{b \in A} |a_1 + b_1, \ldots, a_n + b_n\rangle \langle b_1, \ldots, b_n|$$  \hspace{1cm} (1.26)

$$= X_{N_1}(a_1) \otimes \ldots \otimes X_{N_d}(a_d)$$  \hspace{1cm} (1.27)

where $X_{N_i}$ is the regular representation of a cyclic-group

$$X_{N_i}(a) = \sum_{b=0}^{N_i-1} |a + b\rangle \langle b|$$  \hspace{1cm} (1.28)

A similar decomposition holds for the Fourier transform. Define $\omega_{N_i}$ to be the primitive $N_i$-th root of the unity $\omega_{N_i} := e^{\frac{2\pi i}{N_i}}$. Substituting $\chi_a(b) := \omega_{N_1}^{a_1 b_1} \cdots \omega_{N_d}^{a_d b_d}$ in equation 1.19 one obtains the following equality:

$$F_A = \sum_{a, b \in A} \omega_{N_1}^{a_1 b_1} \cdots \omega_{N_d}^{a_d b_d} |a_1, \ldots, a_n\rangle \langle b_1, \ldots, b_n|$$  \hspace{1cm} (1.29)

$$= F_{N_1} \otimes \ldots \otimes F_{N_d}$$  \hspace{1cm} (1.30)

Where we $F_{N_i}$ is the quantum Fourier transform of each cyclic group:

$$F_{N_i} := \sum_{a, b=0}^{N_i-1} \omega_{N_i}^{ab} |a\rangle \langle b|$$  \hspace{1cm} (1.31)

This decomposition formula shows that to implement any quantum Fourier transform for a finite abelian group efficiently one just needs to be able to implement the cyclic $F_N$. We do not show how to implement this last unitary, since a rigorous analysis several pages. In [10] it is explained how this problem reduces to computing $F_{2^n}$ and $F_N$ with $N$ odd (originally proved in [11]), and then they show how to compute this circuits explicitly. An alternative way is to use $F_{2^n}$ to compute more general $F_N$ using the phase estimation algorithm [25].
CHAPTER 1. FOURIER TRANSFORMS AS UNITARY GATES
Chapter 2

The Hidden Subgroup Problem

Summary

In this section we will review the hidden subgroup problem (HSP), a pure mathematical problem defined over finite groups but also one of most-studied topics of research in quantum computation. A main motivation to study this problem is the fact that the hidden subgroup problem defined over abelian groups can be solved efficiently on a quantum computer, whereas classically it is believed to be an intractable problem. In addition, this problem subsumes many historically important quantum algorithms as special cases, among them Simon’s algorithm [1] and Shor’s algorithm [2], the first known examples of exponential quantum speed-ups.

Regarding the non-abelian HSP, no quantum algorithm has been discovered to solve this problem in full generality. However there is considerable interest on studying its computational complexity, at least, for two particular instances: the symmetric group, and the dihedral group. An algorithm for the former problem would solve graph isomorphism, whereas a solution for the dihedral HSP would give new algorithms for certain lattice problems [15].

In connection with the topic of the present thesis, it turns out that the known quantum algorithms that efficiently solve instances of the hidden subgroup problem (HSP) are very related to Fourier sampling techniques. In particular, abelian quantum Fourier transforms are they key-ingredient to solve the abelian-HSP. In the non-abelian case, techniques based on non-abelian quantum Fourier transforms have had partial success: although there are negative results showing that Fourier sampling seems not to be enough to solve the symmetric HSP [27, 28, 29], these techniques transforms have been used to solve simpler instances of the non-abelian HSP. In particular, several algorithms to solve the HSP for certain families of semi-direct products of abelian groups have been found.

The structure of this chapter is the following. In section 2.1 we will define the hidden subgroup problem and review some important facts about its classical and quantum computational complexity. In section 2.2 we will show that the abelian HSP can be solved for any abelian group making using the abelian quantum Fourier transform and review some of its applications. In section 2.3 we will talk about the non-abelian HSP focusing on semi-direct products of abelian groups. This family shares some similarities with abelian groups but includes hard-instances of the non-abelian HSP such as the Dihedral HSP. In 2.3.3 we will talk about non-abelian Fourier sampling techniques used in quantum algorithms to solve the hidden subgroup problem. In section 2.4 we summarise the relation between the HSP, Fourier sampling and this work.
CHAPTER 2. THE HIDDEN SUBGROUP PROBLEM

2.1 Definition

We introduce now the hidden subgroup problem of a finite group \( G \). Prior to the formal definition consider that we were given a function \( f : G \rightarrow S \) from the group \( G \) to a finite set \( S \) and that this function has the following properties

1. There exists a group \( H \) of \( G \) such that the function fulfills \( f(g_1 h) = f(g_1) \) for all elements \( h \in H \), i.e. the function returns the same value \( f(g_1) \) for all the elements of the coset \( g_1 H \).

2. The function ‘assigns’ different values to different cosets:

\[
 f(g_1) = f(g_2) \quad \text{if and only if} \quad g_1 H = g_2 H
\]  

(2.1)

If a function \( f \) fulfills the above properties we say that \( f \) hides the subgroup \( H \), and that \( H \) is a hidden subgroup. Then, the hidden subgroup problem of the finite group \( G \) is defined as follows.

**Definition 2.1.1** (The Hidden Subgroup Problem). Given a finite group \( G \) and a black-box computing the function \( f : G \rightarrow S \) with a hidden subgroup \( H \), find a set of generators of the hidden subgroup \( H \) using queries of the function \( f \).

Although we used left-cosets in the definition, there is no difference in using right cosets in the definition \[25\]. The hidden subgroup problem is interesting in quantum computation since one can show rigorously that the number of calls of the function \( f \) one must use in order to have enough information to be able to discover the hidden subgroup is exponentially smaller if one uses a quantum computer to try to solve the problem. More rigorously, it is proven in \[42\] that a quantum computer one must do \( O(\text{polylog}|G|) \) queries to the function \( f \) in order to have enough information to be able to compute the group, which scales efficiently. However, in the classical case the number of required queries can grow exponentially \[25\]:

**Theorem 2.1.2.** Suppose that \( G \) has \( N \) different subgroups whose only common element is the identity. Then a classical computer must make \( \Omega(\sqrt{N}) \) queries to solve the HSP.

And there are groups with an exponentially big number of subgroups satisfying the condition of the theorem. Note that although these results show that quantum computers seem to have a considerable natural advantage to solve the HSP they do not guarantee the existence of efficient quantum algorithms to solve this problem and, moreover, they do not exclude the existence of particular groups for which the HSP can be solved classically. In fact, the HSP can be solved classically for groups like \( \mathbb{Z}_p \) or \( \mathbb{Z}_{2^n} \) \[25\]. On the other hand, to solve the HSP for groups like \( \mathbb{Z}_2 \) or \( \mathbb{Z}_N \) a classical computer needs to perform an intractable number of queries \[25\]. In the next section that for any abelian group this problem can be solved efficiently with a quantum algorithm.

2.2 The abelian HSP

In this section we consider the abelian hidden subgroup problem, i.e., the HSP for a group with is commutative. Surprisingly, many already-known quantum algorithms can be re-interpreted as solutions of particular instances of the HSP. In particular, Simon’s algorithm solves a HSP over the abelian group \( \mathbb{Z}_2^n \) and Shor’s algorithm to compute discrete logarithms solves the HSP over the abelian group \( \mathbb{Z}_N \times \mathbb{Z}_N \) \[9, 21, 25\]. All these quantum algorithms solve problems that are believed to be intractable for a classical computer and are examples of exponential quantum speed-ups. A complete list of problems reducible to the abelian-HSP can be consulted in \[3\].

It turns out that any hidden subgroup problem defined over an abelian group \( A \) can be solved efficiently in a quantum computer making crucial use the quantum Fourier transform \( F_A \). We make now a small review of the standard quantum algorithm to solve the abelian-HSP, which can be seen as a generalisation of the phase estimation procedure saw in section 1.1.1.
2.2. THE ABELIAN HSP

2.2.1 The orthogonal subgroup $H^\perp$

The quantum algorithm to solve the abelian-HSP exploits certain dualities between abelian QFTs and subgroups which can be understood in an illustrative way through the study of orthogonal subgroups. We define this group-theoretical concept.

**Definition 2.2.1.** For any subgroup $H$ of an abelian group $A$ the subset of elements of the group $H^\perp$ defined as

$$H^\perp = \{ g \in G \mid \chi_g(h) = 1 \text{ for all } h \in H \}$$

is a subgroup of $A$ known as the orthogonal subgroup of $H$.

It is easy to see that $H^\perp$ is a subgroup since the neutral element belongs to the group $\chi_0(h) = 1$ and given two elements $g_1, g_2$ in $H^\perp$ the difference also belongs to the group using the linearity of the character functions (cf. 1.5.1) which implies $\chi_{g_1 - g_2}(h) = \chi_{g_1}(h)\chi_{g_2}^*(h) = 1$.

**Properties of the orthogonal subgroup**

Now we provide a list of useful properties of $H^\perp$ without proofs. For mathematical details [10] can be consulted

1. The orthogonal of the orthogonal is again the orthogonal.

$$H^{\perp \perp} = H$$

2. The orthogonal $H^\perp$ is isomorphic to the factor group $G/H$.

$$H^\perp \cong G/H \implies |H^\perp| = \frac{|G|}{|H|}$$

The first property tells us that orthogonal subgroups are dual. The second property shows that the order of the orthogonal subgroup is inversely proportional to the order of the original group.

2.2.2 Solving the abelian-HSP via Fourier sampling

Now, we give a general procedure to solve the hidden subgroup problem for any abelian group $A$. For simplicity, we assume that the abelian group is presented to us as a direct-product of cyclic factors $A := \mathbb{Z}_{N_1} \times \ldots \times \mathbb{Z}_{N_d}$ (if the group is not presented in this way there is an efficient quantum algorithm to compute a cyclic-factor decomposition [39]). If we are given a black-box which implements a function $f : A \rightarrow S$ that hides a subgroup $H$, then the quantum-algorithm to solve the abelian hidden subgroup problem is the following (we follow [25]).

1. We begin with a uniform superposition over the group elements and an extra register initialised to zero.

$$|A\rangle := \frac{1}{\sqrt{|A|}} \sum_{a \in A} |a\rangle|0\rangle$$

2. Then we compute the function given by the problem in an extra register.

$$|A\rangle := \frac{1}{\sqrt{|A|}} \sum_{a \in A} |a, f(a)\rangle$$

3. We measure and discard the second register. Since the function is uniform over cosets of $A$ we get with uniform probability a state of the form

$$|a_1 + H\rangle = \frac{1}{\sqrt{|H|}} \sum_{h \in H} |a_1 + h\rangle$$
4. Now we apply the abelian quantum Fourier transform of $A$ from equation (1.19).

$$|a_1 + H⟩ := F_A[a_1 + H]$$

Now consider the sum $\sum_{h \in H} \chi_g(h)$. Since character function of $A$ is an irreducible representation of $H$ the orthogonality rule of the characters (1.16) implies that $\sum_{h \in H} \chi_g(h) = |H| \delta_{g, \text{id}}$, or in words, the sum vanishes unless $\chi_g$ is the trivial representation of $H$. This can happen if and only if $\chi_g(h) = 1$ for every element in $h \in H$, therefore, the sum is not zero only for the elements $g$ which belong to the orthogonal subgroup $H^\perp$. Hence, the final state is a uniform superposition over the orthogonal group $H^\perp$ of the hidden subgroup $H$.

$$|a_1 + H⟩ = \frac{1}{\sqrt{|H^\perp|}} \sum_{g \in H^\perp} \chi_g(a_1)|g⟩$$

5. Finally, we measure this state in the computational basis obtaining random elements from the orthogonal group $g \in H^\perp$. It can be shown that sampling $O(\log |A|)$ elements of the orthogonal subgroup gives a generating set of $H^\perp$ with exponential probability (cf. for instance [40]). Finally, given a generating set of any subgroup $H'$ there is an efficient classical algorithm to compute a set of generators of its orthogonal subgroup $H'^{\perp}$ which we explain in theorem 5.2.2. Applying this to $H^\perp$ we find a set of generators of $H^\perp \perp = H$, which is a generating-set of the hidden subgroup, we are done.

2.3 Non-abelian HSP

In this section we will give a concise review of the state of the art of the non-abelian hidden subgroup problem. Since there are negative results which suggest that the non-abelian HSP in full generality is could be a much-harder problem than its abelian counterpart [27, 28, 29], in this work we restrict to a particular class of non-abelian groups known as semi-direct products $H \rtimes A$. In this section we survey what it is known about the hidden subgroup problem for semi-direct groups and the relationship between the semi-direct HSP and Fourier sampling techniques, which is the main subject we study in this thesis.

2.3.1 The semi-direct HSP in Quantum Computation

Colloquially speaking semi-direct products of the form $H \rtimes A$ are generalisations of direct products $H \times A$ using a non-commutative operation instead of the addition (for a rigorous definition consult chapter 4). Intuitively, semi-direct products of two abelian groups resemble direct-products ‘in structure’, making this class an interesting candidate to study the non-abelian HSP. Although this family of groups may seem a mathematical artifice at first look, there are examples of groups inside this class which are interesting in quantum computation and information, among them the aforementioned Dihedral Group and the Pauli matrices.

1. Perhaps one of the best-known non-abelian finite groups, the Dihedral Group is a semi-direct product with a rather naive structure: it is a combination of two abelian cyclic groups $D_N := \mathbb{Z}_2 \rtimes \mathbb{Z}_N$. In spite of its simplicity, there is considerable interest in understanding the quantum computational complexity of the dihedral-HSP due to a popular reduction that shows that if we found a solution to this problem it would give efficient quantum algorithms for certain lattice problems of interest [13]. The best available algorithm to solve the dihedral-HSP is not efficient and runs in quantum sub-exponential time [10, 13].
2. Other famous semi-direct products in quantum information are the family of groups $\mathbb{Z}_p \ltimes \mathbb{Z}_r^p$ with $p$ a prime number and $r$ fixed, for which efficient quantum algorithms to solve the HSP were discovered in [43, 18]. When $r = 2$, this group is isomorphic to the group Pauli matrices over $\mathbb{Z}_p$, also known as the Heisenberg group. Pauli matrices have applications in several branches of quantum information such as quantum error correction [43], measurement-based quantum computing [45], classical simulations (section 3.3.1) and quantum many-body systems [46].

2.3.2 State of the art of the semi-direct HSP

A big effort has been carried out in the last decade to study the HSP over semi-direct groups similar to the Dihedral Group, with reasonable motivations. First, the quantum complexity of the non-abelian HSP is an interesting problem itself whose study could give insight about the computational capabilities and limitations of quantum computers. Second, this research can yield new ideas for the design of quantum algorithms or highlight unexplored applications of already existing ones.

Slow but constant progress on this field has been achieved over the last decade and several efficient quantum algorithms solving different instances of the semi-direct product HSP have been found. These algorithms illustrate how quantum exponential speed-up can be achieved over classical computation. Throughout the work we will study some particular cases. For further reference the (slightly out-dated) review [10, 45] surveys several of these investigations; [43], [18] and [47] contain some more recent results. The website [3] maintains an up-to-date list with all known relevant quantum algorithms, including those for the non-abelian HSP.

2.3.3 Weak and Strong Fourier sampling

We end this section explaining weak and strong Fourier sampling, two generalisations of the quantum measurement used in the algorithm to solve the abelian HSP (section 2.2) for non-abelian groups. These measurements are mostly used in the design of quantum algorithms to solve hidden subgroup problems.

Just like in the algorithm of the abelian-HSP (section 2.2) we applied the abelian quantum Fourier transform $F_A$ over an abelian coset state followed by a projective measurement on the computational-basis, in the case of non-abelian groups we could generalise this procedure by applying the non-abelian quantum Fourier transform of a finite group

$$F_G := \sum_{g \in G} \sum_{\sigma \in \hat{G}} \sqrt{|G|} \sum_{i,j \in \mathbb{C}} [\sigma(g)]_{j,i} |i,j,\sigma\rangle \langle g|$$

over non-abelian coset-states and then measure. Non-abelian coset states can be created mimicking the procedure used in the abelian case: create a uniform superposition over all group elements, apply the unitary that implements the input function $f$ hiding a given subgroup $H$, then measure and discard the register were the function $f$ acts. This procedure creates states of the form

$$|gH\rangle := \frac{1}{\sqrt{|G|}} \sum_{h \in H} |gh\rangle$$

(2.12)

The goal in the hidden subgroup problem is to design a quantum measurement that allows us to extract the subgroup $H$ from these coset states. Weak and strong Fourier sampling are two measurement-schemes that are often used in this context.

Weak Fourier sampling

Weak Fourier sampling is a measurement scheme used to extract information about the hidden subgroup over coset states $|gH\rangle$ of a non-abelian group $G$ based on non-abelian Fourier sampling. The procedure is the following.

1. Apply the non-abelian quantum Fourier transform $F_G$ on the state. The gate was defined in equation 1.12.
2. Then measure the label $\sigma$ of a representation. That is, given the state $F_G|gH\rangle$, perform a quantum measurement described by projectors.

$$P_{\sigma} := \sum_{i,j=0}^{d_{\sigma}} |i,j,\sigma\rangle\langle i,j,\sigma|$$ (2.13)

3. Use the labels $\sigma$ measured to try to find the hidden subgroup $H$.

This measurement scheme is not enough to solve the hidden subgroup problem but for a rather small class of subgroups known as Hamiltonian groups [48]. However, weak Fourier sampling has some nice properties that allow to use it as a subroutine to implement more complicated measurements such as Strong Fourier sampling or Optimal Measurements for certain groups [43]. The first one is that measuring the label of a representation in the above procedure can be done without loss of information about the hidden subgroup. The second is that the output distribution over the labels is independent of the choice of basis for the irreducible representations which is used to implement the quantum Fourier transform. Both features are surveyed in [25].

**Strong Fourier sampling**

Strong Fourier sampling is an enhanced version of weak Fourier sampling where one measures not only the label of the irreducible representations but also the part of the Hilbert space corresponding to the matrix coefficients of them.

1. Apply the non-abelian quantum Fourier transform $F_G$ on coset state $|gH\rangle$.
2. Do a projective measurement defined by projectors $P_{i,j,\sigma} := |i,j,\sigma\rangle\langle i,j,\sigma|$.
3. Use the information obtained $(i,j,\sigma)$ to try to find the hidden subgroup problem.

By definition, strong Fourier sampling is more powerful than weak Fourier sampling. However, the output distribution of the outcomes depends on the basis of the irreducibles of the irreps, a degree of freedom which has to be taken in account in applications. It is not known whether proper choices of basis can significantly improve Strong Fourier sampling [25].

### 2.4 Relation to this work

In this thesis, we will study the classical simulation of quantum Fourier transforms over semi-direct products of the form $\mathbb{Z}_p \rtimes A$ where $A$ is an arbitrary abelian group, $\mathbb{Z}_p$ is the abelian cyclic group of prime-order $p$ and $\varphi$ an automorphism of the abelian group $A$ of order $p$. We find several motivations to study the simulation of non-abelian Fourier transforms over this class of groups.

First, in connection with the known-results showing that abelian quantum Fourier transforms can be efficiently simulated classically (cf. [11, 12, 13] and chapter 3), the feasibility of simulating non-abelian quantum Fourier transforms efficiently arises as a natural question which, to our best knowledge, has not been answered yet. In this context, it makes sense to choose first a “simple” family of non-abelian groups, like $\mathbb{Z}_p \rtimes \varphi A$. We highlight that this class is realistic but not naive, since there is no known quantum algorithm to solve the HSP for the whole class. Some interesting examples of groups inside this family are the Dihedral Group and the group of Pauli matrices we mentioned in this chapter, as well as other instances studied in the field of quantum algorithms [17, 43, 18, 19, 20].

The second motivation to study quantum Fourier transforms for groups $\mathbb{Z}_p \rtimes \varphi A$ is that we know that they help to solve the HSP over these problems something which is not true in the general non-abelian case [27, 28, 29]. Experience shows that all currently known quantum-algorithm solving instances of the HSP for the class $\mathbb{Z}_p \rtimes \varphi A$ make use of Fourier sampling techniques. Interesting examples in the literature are the wreath products $\mathbb{Z}_2 \rtimes \mathbb{Z}_2^n$ [17], a family of groups $\mathbb{Z}_2 \rtimes \mathbb{Z}_p^n$ from [19], semi-direct groups $\mathbb{Z}_p \rtimes \mathbb{Z}_p^n$ from [20], the class of groups $\mathbb{Z}_p \rtimes \mathbb{Z}_r^+$ for fixed $r$ which includes the Pauli group [43]. In spite of being inefficient, the quantum algorithm to solve the dihedral-HSP is also based in Fourier sampling [15, 15].
Chapter 3

Classical simulation of quantum computers

Summary

In this chapter we give an introduction to classical simulation of quantum computations, which is an approach to explore the differences between quantum and classical computational power based on the identification of classes of quantum circuits that can be efficiently simulated in a classical computer. Unlike the quantum algorithms seen in chapter 2 a quantum process that can be efficiently simulated classically can not offer an exponential speed-up over classical computation. The investigation of such quantum processes helps us to understand the key features of quantum mechanics that are responsible for quantum computational power.

The structure of the chapter is the following. In section 3.1 we give some introductory motivation to study classical simulations and survey some interesting classes of quantum circuits that can be simulated. In section 3.2 we introduce weak simulations, which is the notion of classical of simulations we will use in this thesis. In section 3.3 we talk about previous results about classical simulation of abelian quantum Fourier transforms related to the present work. In section 3.4 we will comment possible applications.

Convention: during the rest of this work ‘classical simulation’ will be synonym to ‘efficient classical simulation’.

3.1 Introduction

Nowadays, a considerably big number of quantum algorithms are known, several of them showing exponential quantum speed-ups [3]. Yet, understanding the fundamental questions “what is the power of quantum computers compared to classical ones?” and “which quantum circuits are useful to achieve these effects?” is still a major challenge of the field of quantum computation.

An approach to tackle these problems is studying classes of quantum computations that can be efficiently simulated in a classical computer and, hence, can not offer exponential-savings of computational resources. These investigations can improve our comprehension of the essential features of quantum mechanics that can help to find more efficient solutions to computational problems or indicate where to look for new algorithmic primitives. Work during the last decade has shown that the lack of certain types of entanglement [5, 49, 50, 51, 52] during a computation allows an efficient classical simulation. However,

\[1\text{Quantified in terms of suitable entanglement measurements.}\]
 CHAPTER 3. CLASSICAL SIMULATION OF QUANTUM COMPUTERS

exotic families of quantum circuits that allow to create large degrees of entanglement, quantum interference and superpositions can also be efficiently simulated classically among them, the Gottesman-Knill theorem circuits \[4, 53, 54\] and matchgate-circuits \[6, 55, 56, 57\]. Current understanding suggests that several quantum features such as entanglement and interference must be present in a quantum circuit in order that exponential speed-ups can show-up.

3.2 Weak simulations

There are different notions of classical simulations that have been used in the literature which we respectively call strong and weak simulations. In this work we will focus on the latter ones. In this section we will give a non-technical survey on the known differences between both. For details, \[8\] is a clear reference on topic. Now, let us briefly remind the main steps of a quantum computation \[7\]:

1. A \(n\)-qubit register is initialised in a certain quantum state which is the input of the computation.
2. A sequence of unitary gates (a quantum circuit) is applied.
3. At the end, the qubits are measured and we obtain a bit-string as output. The bit string is usually not determined and comes from a probability distribution.

Regarding the above procedure, in a strong simulation of a quantum computation, the goal typically is to compute with high precision the final measurement probabilities of the outcomes (or the expectation value of some operator) using an efficient classical algorithm. In a weak simulation the objective is to find an efficient classical procedure that allows to sample from the resulting output probability distribution.

As it is discussed in \[7, 8\], since the nature of quantum is intrinsically probabilistic, the weak notion of classical simulation seems a priori a more natural approach in order to make comparisons between classical and quantum computation. Besides, there are rigorous established observations showing that the use of strong simulations to compare quantum and classical computational power can lead to unrealistic scenarios. In particular, there are simple examples of quantum circuits that can be simulated with weak simulations whereas a strong simulations turns out to be intractable\[7, 8\].

Conventions: throughout this thesis we will study quantum computations that can be simulated in the weak sense, i.e. we will look at some quantum circuits acting on some initial quantum states and consider some measurements on the output state. If we can sample from the output distribution of the quantum measurement with an efficient classical algorithm then we say that we can classically simulate the quantum computation.

3.3 Simulation of abelian QFTs

In relation to our work, we will now comment preceding results showing that the classical simulation of abelian quantum Fourier transforms is possible. These results are obviously related to our work, since we have considered classical simulations in the non-abelian case.

3.3.1 The Gottesman-Knill theorem

The Gottesman-Knill theorem is one of the oldest results about classical simulations of quantum circuits, which presents a flexible class of quantum circuits whose action on computational-basis states followed by a measurement can be efficiently simulated on a classical computer. Although the theorem implies that the family of quantum processes being considered is not sufficient to perform universal quantum computation, they can create pure quantum phenomena such entanglement and quantum interference. The unitary gates describing these quantum circuits have been deeply studied in Quantum Information and Computation and they are usually referred as ‘the Clifford group’. We present an equivalent version of the original theorem \[4\].

\[2\]In the circuit model, which is a universal model of quantum computation.
Theorem 3.3.1 (Gottesman-Knill theorem). Given a poly-size quantum circuit $C$ acting on $n$-qubits, which only involves the following type of gates: Pauli gates, Hadamard gates, phase gates and CNOT gates. Then the action of this circuit on an original computational-basis state $|x\rangle$ followed by a measurement on the computational-basis can be efficiently simulated on a classical computer, i.e. there is a classical algorithm to sample from the probability distribution of measurement-outcomes using $\text{poly}(n)$ time and memory resources.

**Proof:** We sketch the main ideas of the proof and refer to [4],[9] for some details. The key observation to handle this problem is that the original input states can be described in terms of Pauli operators, following a ‘Heisenberg picture’ style. We remind the definition of the so-called Pauli gates:

$$I := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y := \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix}, \quad Z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$  \hspace{1cm} (3.1)

Where $Y$ can be written as $iZX$ and we rarely consider it individually. An arbitrary Pauli operator is any unitary gate of the form $\sigma_1 \otimes \sigma_2 \otimes \ldots \otimes \sigma_n$ where each $\sigma_i$ belongs to the group generated by the Pauli matrices. Now, for simplicity, imagine we are given an initial $n$-qubit quantum-state $|0\rangle = |0_1 0_2 \ldots 0_n\rangle$, then this state can be described unique description in terms of Pauli Operators as: the unique-common-eigenstate with eigenvalue $+1$ of the group generated by the gates:

$$Z_i := I_1 \otimes \ldots \otimes I_{i-1} \otimes Z_i \otimes I_{i+1} \otimes \ldots \otimes I_n, \hspace{1cm} (3.2)$$

Where gate $Z_i$ applies a $Z$ gate on the $i$th qubit system and the identity elsewhere. It is also evident that there are no Pauli operators satisfying this property but the ones contained in the group of matrices generated by the $Z_i$s.

Now, imagine we have a circuit $C$ of the form explained by the theorem over the initial input, obtaining $|\psi_C\rangle := C|0\rangle$. Simulating a measurement on the computational basis after applying this circuit is equivalent to sample from the probability distribution $\pi_x := |\langle x|\psi_C\rangle|^2$, where the new state $|\psi_C\rangle$ is uniquely described as the common $+1$ eigenstate of a new set of operators $U_i := CZ_iC^\dagger$. Now, there are two important properties of this state which simplify the sampling problem:

1. The operators $U_i := CZ_iC^\dagger$ are again Pauli operators.

2. If we measure a single qubit in the computational-basis, the state after the measurement is again described as the common $+1$ eigenstate of a new set of Pauli operators. Therefore, we can simulate a whole computational-basis measurement recursively if we can do it only for one qubit.

For the first property, we can decompose the circuit $C$ as a product of the local one and two qubit gates given in the statement of the theorem, so it is enough to prove that all them satisfy this condition. Apart from Pauli gates, the other one-qubit unitaries we care about are the Hadamard gate $H$ and the phase gate $P$, defined as:

$$H := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad P := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix},$$  \hspace{1cm} (3.3)

And the gate CNOT is defined as the two-qubit unitary that applies the Pauli $X$ gate on one qubit conditioned on the value of the other one:

$$\text{CNOT}|x_1, x_2\rangle = |x_1, x_1 \oplus x_2\rangle$$  \hspace{1cm} (3.4)

In [4] it is shown that these Clifford gates transform Pauli operator as follows:

<table>
<thead>
<tr>
<th>Hadamard, $H$</th>
<th>Phase, $P$</th>
<th>CNOT $X \otimes I \rightarrow X \otimes X$</th>
<th>CNOT $Z \otimes I \rightarrow Z \otimes I$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X \rightarrow Z$</td>
<td>$X \rightarrow Y$</td>
<td>$X \otimes I \rightarrow X \otimes X$</td>
<td>$Z \otimes I \rightarrow Z \otimes I$</td>
</tr>
<tr>
<td>$Z \rightarrow X$</td>
<td>$Z \rightarrow Z$</td>
<td>$I \otimes X \rightarrow I \otimes X$</td>
<td>$I \otimes Z \rightarrow Z \otimes Z$</td>
</tr>
</tbody>
</table>

---

3Here the labels are written in binary representation, hence, $x$ is a $n$-bit string.
This proves the first property. To prove the second we almost will follow the proof of \cite{9}. Now we try to measure \(w.l.o.g\) the first qubit of the final state, or equivalently to measure the operator \(g := Z \otimes I_{n-1}\). As it is a Pauli operator, it has to either commute with all the \(U_i\) or anti-commute with some of them.

First, if it commutes it follows that \(g |\psi_C\rangle\) is a common \(+1\) eigenstate of all \(U_i\), hence proportional to \(|\psi_C\rangle\). This implies that so \(|\psi_C\rangle\) is an eigenstate of \(Z\) so it must hold \(g |\psi_C\rangle = \pm |\psi_C\rangle\), as the unique eigenvalues of \(Z\) are \(\pm 1\). This shows that one would measure 0 or 1 with probability one; we only have to determine which one. The last equation shows that either \(g\) or \(-g\) belongs to the stabiliser group generated by the \(U_i\), which is actually a commutative group. Then \(d = CgC\) is diagonal and proportional to a product of gates \(Z\):

\[
\pm d = Z_1^{a_1} \ldots Z_n^{a_n} \tag{3.5}
\]

Where the equations for the row \(x\) (this an n bit string) is \(\pm d_x = (-1)^{a_x} d_x\) where \(d_x = (-1)^{b_x}\) for a constant \(b_x\) which we can compute. Then we obtain two systems of linear modular equations. It can be checked directly which one has a solution by turning it into a normal linear system and using gaussian elimination. Alternatively, the algorithm we give in the proof of theorem \[7.2.2\] to solve systems of modular equations can be applied.

Second, if it anti-commutes with one or more \(U_i\), it is possible to choose new operators \(V_i\) such that \(g\) only anti-commutes with the first of them (multiplying pairs of anti-commuting ones), so we do this case. \(g\) has two eigenspaces with \(\pm 1\) eigenvalues, so the projected state of the system after a measurement will be of the form \((I \pm g) |\psi_C\rangle / \sqrt{2}\) depending of the measurement outcome. The new operators describing this state are \(\{\pm g, V_2, \ldots, V_n\}\). Finally, it is easy to see how to sample, since writing the explicit formulas for the probability of each outcome one obtains \(p(+1) = p(-1)\) (cf. \cite{9}), which simplifies the process to generating 0 or 1 randomly.

Finally, it is immediate that we can start with any input state \(|b\rangle\) instead of \(|0\rangle\), as \(|b\rangle := X_1^{b_1} \otimes \ldots \otimes X_n^{b_n} |0\rangle\) and the gates applied form a Clifford circuit which can be appended to \(C\). \(\blacksquare\)

An implication of Gottesman-Knill theorem is that circuits containing Hadamard gates, which is the quantum Fourier transform of the group \(Z_2\), can be efficiently simulated classically.

### 3.3.2 Simulation of cyclic quantum Fourier transforms

As we explained in the introduction of this thesis, relatively recent works have shown that the abelian quantum Fourier transform \(F_{2^n}\) of the cyclic group \(Z_{2^n}\) can be efficiently simulated classically when it acts on certain families of initial quantum-states \[11\] \[12\] \[13\] and it is followed by a projective measurement. These states can be product-states, quantum-states generated by log-depth circuits with limited-range interactions and certain classes of MPS and graph states.

This thesis is very related to these preceding works. In our set-up, we consider non-abelian Fourier transforms and non-abelian Fourier sampling measurements. We study the action of these quantum processes on arbitrary coset-states of the non-abelian groups \(Z_p \ltimes \varphi A\), including as particular cases: computational-basis states, abelian-subgroup cosets, normal-subgroup cosets and the coset-states for which the HSP over \(Z_p \ltimes \varphi A\) is more difficult to solve \[43\].

It is important to notice that the results of these works do not imply that quantum algorithms with exponential quantum speed-ups using quantum Fourier transforms (like Shor’s algorithm) can be efficiently simulated classically. However, these observations tell us that abelian and non-abelian quantum Fourier transforms, that are relatively complex unitaries can be efficiently simulated classically.

### 3.4 Applications

So far, we have not discussed “practical” applications in quantum information of the classical simulation results we have commented. In connection with the Gottesman-Knill theorem, the class of circuits described in this result are usually called stabilizer states and have applications in quantum error correction \[44\]. However, in the case of classical simulation of abelian quantum Fourier transforms, clear applications have still not been found \[14\] contains a short discussion on this topic).
3.4. APPLICATIONS

We finish the chapter commenting that it is possible to extend some of the results commented above combining them with recent discoveries from [8] to show that the measurement of some interesting physical observables applied to the output states of Clifford circuits or Fourier transforms can be estimated with efficient classical algorithms. In particular, in [8] it is shown that the ability to sample classically from a quantum state and to compute its computational-basis coefficients can be used to simulate more complicated observables that projective measurements. Rigorously, the authors define a class of states called “computationally tractable states”.

**Definition 3.4.1.** An \( n \)-qubit quantum state \( |\psi\rangle \) is “computationally tractable” if the following conditions hold:

(a) It is possible to sample in \( \text{poly}(n) \) time with classical means from the probability distribution \( \text{Prob}(x) = |\langle x |\psi\rangle|^2 \) where \( g \) belongs to the finite group \( G \).

(b) For any \( n \)-bit string \( x \) the coefficient \( \langle x |\psi_G\rangle \) can be computed in \( \text{poly}(n) \) time on a classical computer.

This definition subsumes several families of quantum states which are used for quantum-informational tasks and of very different nature. Some of them have been already considered in this chapter:

- Matrix Product States of polynomial bond dimension [8].
- Stabilizer states [4, 54], i.e. the ones considered in the Gottesman-Knill theorem in section 3.3.1.
- Related to section 3.3, the states state obtained after applying the quantum Fourier Transform (over the integers modulo \( 2^n \)) to an arbitrary product state [12, 11, 13].

Perhaps surprisingly, for all these states, regardless of their different nature it can be shown that several examples of quantum processes that are described by sparse operators can be simulated classically including certain types of measurements used in quantum information and condensed matter physics [8]. We just cite some examples taken from [8].

- Any local-measurement acting at most in \( \log(n) \) qubits.

- The measurement of an observable which decomposes as a sum of \( \text{poly}(n) \) sparse observables, in particular, \( k \)-local Hamiltonians with \( k = \log(n) \).

- Measuring an operator which is tensor products of Pauli Gates \( (X, Y, Z) \) which may or not be preceded by the action of a Clifford circuit (those considered in the Gottesman-Knill theorem [4]).
Part II

Group Theory
Chapter 4

Semi-direct products

Summary

In this chapter introduce a simple class of non-abelian groups, the semi-direct products of the form \( \mathbb{Z}_p \rtimes \varphi \mathcal{A} \). We define this groups and show some of their most important properties relevant to us. Convention. Throughout the rest of the work whenever we talk about ‘semi-direct product groups’ we implicitly refer to ‘semi-direct products of abelian groups’.

4.1 Definition

**Definition 4.1.1.** For any abelian group \( \mathcal{A} \) with an automorphism \( \varphi \) of primer order \( p \), a semidirect product group \( \mathbb{Z}_p \rtimes \varphi \mathcal{A} \) can be constructed as the set of elements \( (m, a) \in \mathbb{Z}_p \times \mathcal{A} \) with one of the two following multiplication rules:

\[
(m, a) \circ (n, b) = (m, a) + \varphi^m(n, b) \\
(m, a) \bullet (n, b) = \varphi^n(m, a) + (n, b)
\]

Where \( \varphi \) is acts only on the second register \( \varphi(m, a) := (m, \varphi(a)) \). The inverse of an element \( (m, a) \) is \( (-m, -\varphi^{-m}(a)) \). The order of this group is \( |G| = p|\mathcal{A}| \).

Both possible definitions above have been used in the field of Quantum Algorithms. In this work we choose to use the “white” operation and from now we write simply \( (m, a)(n, b) = (m, a) \circ (n, b) \) without showing the multiplication symbol. This is a matter of taste since it can be shown that they define isomorphic groups (see next section). In practice, we prefer the white multiplication since it sometimes leads to simpler formulas.

4.1.1 Opposite group

Now, we want to highlight that both the ‘black’ and the ‘white’ groups are isomorphic to each other, a remark which is often skipped. To show it, it is enough to show that both groups are opposite, a group-theoretical concept which we define now.

**Definition 4.1.2.** Given a group \( G = (X, \bullet) \) (set plus operation) its Opposite Group \( G^{op} \) is defined as \( G^{op} := (X, \circ) \) where \( \circ \) is called the “opposite” operation of \( \bullet \)

\[
g \circ h = h \bullet g \quad \forall g, h \in G
\]

\( G^{op} \) and \( G \) are isomorphic via inversion: \( \text{Inv}(g) \rightarrow g^{-1} \)

It can be easily checked that the “white” and the “black” semidirect groups in definition 4.1.1 have opposite operations and, hence, they are opposite and isomorphic.

35
CHAPTER 4. SEMI-DIRECT PRODUCTS

Remark
Note that the definition of semi-direct product 4.1.1 can be generalised replacing the group \( \mathbb{Z}_p \) with an abelian group of automorphisms \( H := \langle \varphi_1, \ldots, \varphi_k \rangle \cong \mathbb{Z}_{r_1} \times \ldots \times \mathbb{Z}_{r_k} \), where \( r_i \) is the order of each generator of the automorphism. Although our work focuses on the class \( \mathbb{Z}_p \ltimes \varphi A \), groups of the form \( H \ltimes A \) appear in the literature and in some of theorems we use in appendix A.

4.2 Arithmetic operations
The following definitions will be useful to describe arithmetic operations on this groups such as multiplication and exponentiation and therefore, also to write arbitrary elements of a subgroup in terms of their generating-sets.

Definition 4.2.1. We define a family of functions \( \{ \Phi_m \} \) from \( \mathbb{Z}_p \times A \) to \( A \) indexed by a number \( m \in \mathbb{Z}_p \):

\[
\Phi_m^{(n)}(a) := \sum_{i=0}^{n-1} \varphi^{im}(a)
\] (4.4)

Using these functions, the nth power of an element \((m,a)\) can be written as

\[
(m,a)^n = (mn, a + \varphi^m(a) + \varphi^{2m}(a) + \ldots + \varphi^{(n-1)m}(a))
\] (4.5)

\[
= (mn, \Phi_m^{(n)}(a))
\] (4.6)

Above, realise that in \( \Phi_m^{(n)}(a) \) the number \( n \) is not a power but an input-parameter of the function. Also, \( n \) matches the number of elements that are summed in the second register of equation (4.5).

4.3 Subgroups
Throughout the work we will use the following notation and definitions to describe the subgroups \( H \) of the \( \mathbb{Z}_p \ltimes \varphi A \) and its cosets \((m,a)H\).

Definition 4.3.1. For any subgroup \( H \) of \( \mathbb{Z}_p \ltimes \varphi A \) the group \( \{0\} \times A \) is always an abelian subgroup isomorphic to \( A \) which we will denote just \( A \). Also, we will use the following notation:

- \( H_A := H \cap A \), which is always a subgroup of the abelian group \( A \).
- \( H_A^\perp \) will be the set of elements \((0,g)\) fulfilling \( \chi_g(h) = 1 \) for any \((0,h) \in H_A \) and it is a subgroup of \( A \). Mind that it is the orthogonal of \( H_A \) inside \( A \), not inside the direct group \( \mathbb{Z}_p \ltimes \varphi A \).

A useful lemma proven in [43] completely classifies the subgroups of the semi-direct products \( \mathbb{Z}_p \ltimes \varphi A \). We present a slightly re-organised version of the lemma equivalent in content.

Lemma 4.3.2. Any subgroup \( H \) of a semidirect group \( \mathbb{Z}_p \ltimes \varphi A \) with \( p \) a prime number and \( A \) an arbitrary abelian group belongs to one of the two following classes:

1. \( H = H_A \) and it is a subgroup of \( A \).
2. \( H = \langle (1,d), H_A \rangle \) where:
   
   (a) \( H_A \) is a normal subgroup of \( H \)
   
   (b) \( H_A \) is invariant under the action of the automorphism \( H_A = \varphi(H_A) \)
   
   (c) The quotient group \( H/H_A = \langle (1,d) \rangle \) is cyclic of order \( p \).
4.4. COSETS

Lemma 4.3.2 is a very useful tool which allows to described the subgroups of $\mathbb{Z}_p \rtimes \varphi A$ in terms of subgroups and cosets of $A$. We can use it to prove that any subgroup $H$ of type 2. can be decomposed as a disjoint union of cosets of its abelian subgroup $H_A$.

Corollary 4.3.3. Any subgroup $H$ of $\mathbb{Z}_p \rtimes \varphi A$ not fulfilling $H \neq H_A$ can be decomposed as a disjoint union of $p$ cosets of its abelian subgroup $H_A$.

$$H = \bigcup_{n=0}^{p-1} \{(1, d)^n + H_A\}$$

(4.7)

Proof: The subgroups are those of type 2 in lemma 4.3.2 whose properties imply that $(1, d)H_A = H_A(1, d)$. Therefore $H = \langle (1, d), H_A \rangle$ can be written as $H = \langle (1, d) \rangle H_A$.

4.4. Cosets

Finally, we show that the previous result can be extended to any coset of the group $\mathbb{Z}_p \rtimes \varphi A$ and give a complete classification (box 4.1) of them that will be useful in chapter 7 where we will study the simulation of QFT acting on coset-states of this groups.

Lemma 4.3.2 and corollary 4.3.3 can be used to show that any coset $(m, a)H$ of $\mathbb{Z}_p \rtimes \varphi A$ can be be decomposed as a disjoint union of 1 or $p$ abelian cosets.

1. Case $H = H_A$.

$$H = \bigcup_{n=0}^{p-1} \{ (n, \Phi^n_m (\varphi^m(d))) + H_A \}$$

(4.9)

Where $H' := \varphi^m(H_A)$ is a new abelian subgroup since $\varphi$ is a group automorphism of $A$.

2. Case $H \neq H_A$.

$$H = \bigcup_{n=0}^{p-1} \{ \Phi^n_m (\varphi^m(d)) + H_A \}$$

(4.10)

Box 4.1: Complete classification of the cosets of $\mathbb{Z}_p \rtimes \varphi A$

To prove the second case in box 4.1 we have used the (white) multiplication rule of the group (4.1.1) properties of subgroups of type 2 in lemma 4.3.2 in particular that the subgroup $H_A$ is invariant under the action of the automorphism. With these properties is easy to show:

$$H_A \langle (1, d)^n \rangle = (m, a) + \varphi^m [(n, \Phi^n_m (d)) + H_A]$$

(4.10)

$$H = \bigcup_{n=0}^{p-1} \{ [n, \Phi^n_m (\varphi^m(d))] + H_A \}$$

(4.11)

and the rest follows redefining the variables in the decomposition properly.
CHAPTER 4. SEMI-DIRECT PRODUCTS
Chapter 5
Computational complexity of group-theoretical problems

Summary
In this chapter we present some algorithms to perform certain computational tasks over abelian groups and semi-direct product groups of the form $\mathbb{Z}_p \ltimes \phi A$ and discuss their efficiency. Throughout this thesis we will need these algorithms in order to show how to simulate classically certain quantum measurements. In section 5.1 we introduce some computational complexity conventions. In section 5.2 we give algorithms for abelian groups. In section 5.3 we focus on semi-direct products.

5.1 Conventions

We remind the following conventions that we will use and assume through the whole work. As we did for $G$-states (re-visit the introduction of chapter 1 if necessary) we define an input-size $n := \lceil \log |G| \rceil$ which we will use to measure the efficiency of computational-processes. An algorithm that solves a given problem over a finite group $G$ is said to be efficient if it uses $O(poly(n))$ time and memory resources before stopping. Regarding memory resources, we will call any set whose number of elements is at most $poly(n)$ a ‘poly-size’ set.

Encoding of the group elements
We will always work with abelian groups presented as a decompositions of cyclic-factors, where cyclic groups are chosen to be non-trivial without loss of generality.

$$A = \mathbb{Z}_{N_1} \times \ldots \times \mathbb{Z}_{N_d}$$ (5.1)

We assume that such decomposition is always given to us as an input. The elements of the group will be always presented as tuples of numbers $a := (a_1, \ldots, a_d)$. In this notation, the elements $e_i := (0, \ldots, 1_i, \ldots, 0_d)$ form a generating-set of the group $A := \langle e_1, \ldots, e_d \rangle$ and the element $a := (a_1, \ldots, a_d)$ is written as:

$$a := (a_1, \ldots, a_d) = \sum_{i=1}^{d} a_i e_i$$ (5.2)

We will always describe the subgroups $H$ of $A$ in terms of generating-sets $\langle h_1, \ldots, h_r \rangle$. Generating sets provide an economic way to describe subgroups since any finite group can be described with at

\footnote{Intuitively, a measure of the size of our system.}
most $O(n)$ of generators \([5]\). Since any element can be stored with $O(n)$ bits using the tuple-encoding it follows that any generating-set can be stored using $O(n^2)$ memory and is a poly-size set.

**A final remark.** This encoding is the most common choice for semi-direct in the HSP literature including the dihedral group and several extensions \([17, 43, 19, 20]\). In this work we will never discuss black-box groups or alternative encodings.

### 5.2 Algorithms for abelian groups

First of all it is important to remember that there are efficient classical algorithms to perform modular arithmetic. This topic is treated in most textbooks on algorithmics (a good summary can be accessed online in \([58]\)). For our purposes it is enough to keep the following proposition in mind:

**For any composite number** $N$, **there are efficient classical algorithms to add, multiply and exponentiate modulo** $N$.

As a result, for any abelian group $A = \mathbb{Z}_{N_1} \times \ldots \times \mathbb{Z}_{N_d}$ there are (efficient) classical algorithms to add $a + b$ and multiply $k \cdot a$ inside the group $A$ which consume at most $\text{polylog}|A|$ time and memory resources.

#### 5.2.1 Computing abelian group automorphisms

For a given abelian group $A = \mathbb{Z}_{N_1} \times \ldots \times \mathbb{Z}_{N_d}$ any automorphism $\varphi$ of $A$ has a canonical presentation as a $d \times d$ integer matrix $M_\varphi$. Using that automorphisms are linear functions fulfilling $\varphi(a + b) = \varphi(a) + \varphi(b)$ it follows:

$$a := (a_1, \ldots, a_d) = \sum_{i=1}^{d} a_i e_i \quad \implies \quad \varphi(a) = \sum_{i=1}^{d} \varphi(e_i) a_i \quad (5.3)$$

This formula allows to compute the automorphism function $\varphi(a)$ canonically as the product of a matrix $M_\varphi$ acting on the element $a$ (considered here as a tuple). The matrix $M_\varphi$ is constructed canonically using the action of the automorphism on the canonical generators $e_i$. Define $[M_\varphi]_{i,j} := \varphi(e_j)_i$, then the problem of computing automorphisms reduces to matrix multiplication modulo $A$.

$$\varphi(a) = M_\varphi \cdot a^T \pmod{A} = \begin{pmatrix} \varphi(e_1) a_1 + \varphi(e_2) a_2 + \ldots + \varphi(e_d) a_d &= \text{mod } N_1 \\ \varphi(e_1) a_1 + \varphi(e_2) a_2 + \ldots + \varphi(e_d) a_d &= \text{mod } N_2 \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(e_1) a_1 + \varphi(e_2) a_2 + \ldots + \varphi(e_d) a_d &= \text{mod } N_d \end{pmatrix} \quad (5.4)$$

Where $i$th component $\varphi(a)_i$ of the new element is computed performing additions and multiplications modulo $N_i$. It is a common practice to use this canonical matrices to define the group automorphisms.

Now we prove that group automorphisms can always be computed if one uses this presentation:

**Lemma 5.2.1.** For any abelian group $A = \mathbb{Z}_{N_1} \times \ldots \times \mathbb{Z}_{N_d}$ any group automorphism $\varphi$ can be efficiently computed if the canonical matrix $M_\varphi$ is given as an input.

**Proof:** We show how to that $b := \varphi(a) = M_\varphi a^T$ can be computed efficiently for any general $a = (a_1, \ldots, a_d) \in A$. The cost of computing $b_i = \sum_{j=1}^{d} [M_\varphi]_{i,j} a_j \pmod{N_i}$ using modular arithmetic on $N_i$ is bounded by $O(d \times \text{polylog}|A|)$ thus the matrix multiplication. Finally, we can upper-bound $d$ substituting $N_i \geq 2$ (which holds since no factor is trivial) in the formula of the order of the group $|A| = \prod_{i=1}^{d} N_i$. One obtains the inequality $2^d \leq |A|$ from where $d = O(\log |A|)$. Substituting this bound one obtains that the total cost of the automorphism is $O(\text{polylog}|A|)$.  

\[\blacksquare\]
5.2. ALGORITHMS FOR ABELIAN GROUPS

5.2.2 Advanced problems for abelian groups

In this section we prove how to solve three important problems for abelian groups that we will need later in chapter 7 to show how to simulate quantum Fourier transforms.

Theorem 5.2.2. Given any abelian group $A = \mathbb{Z}_{N_1} \times \ldots \times \mathbb{Z}_{N_d}$ and a subgroup $H \subset A$ with a poly-size generating set $\{h_1, \ldots, h_r\}$, there are efficient classical algorithms to solve the following problems:

(a) Sampling random elements from the subgroup $H$ with equal probability.

(b) Deciding membership of an element $g \in G$ in the subgroup $H$.

(b) Finding a poly-size generating-set of the orthogonal subgroup $H^\perp$.

Proof: We prove (a) first because it is the easiest algorithm. Then (c) first since there is a simple polynomial reduction from (c) to (b).

(a) Sampling from a random subgroup. Just computing $r$ random $k_i$ numbers from 0 to $|A|$ and compute $h = \sum k_i h_i$. We can see that we sample an element with uniform probability as follows. The generating set $\{h_1, \ldots, h_r\}$ defines an inclusion series $H_1 \unlhd H_2 \unlhd \ldots \unlhd H_r = H$ where each group is defined as $H_i := \langle h_1, \ldots, h_i \rangle$. Because the subgroups $(h_{i+1})$ and $H_i$ commute, the subgroup $H_{i+1}$ can be written as $H_{i+1} = (h_{i+1})H_i$. Denote $r_i = |H_{i+1}|/|H_i|$ the order of the factor group $H_{i+1}/H_i$, the group $H_{i+1}$ can be decomposed as a disjoint union of $r_i$ cosets of $H_i$:

$$H_{i+1} = \bigcup_{l=0}^{r_i-1} lh_{i+1} + H_i \quad (5.5)$$

Now imagine an element $g_i \in H_i$ is given to us by a subroutine which samples elements from $H_i$ with uniform probability. If we compute a random number $k_{i+1}$ from 0 to $|A|$ the residue of $k_{i+1}$ modulo $r_i$ will be a uniform number in $\mathbb{Z}_{r_i}$, since the number $r_i$ divides $|A|$ due to Lagrange’s theorem. As a result, the element $k_{i+1} h_{i+1} + g_i$ will be a random element in $H_{i+1}$. Now, to sample from $H$ uniformly we sample from the first group of the inclusion chain $H_1 := \langle h_1 \rangle$. Since this is cyclic we only need to compute, again, a random multiple $k_1 h_1$ with $k_1$ from 0 to $|A|$. This procedure can be used as a subroutine to sample from the next group $H_2$. In this way we can construct an iterative procedure to sample from any subgroup in the inclusion chain, which gives a polynomial-time method to sample uniformly from $H$. The outcomes of the sampling procedure are precisely those elements of the form $h = \sum k_i h_i$ giving at the beginning.

(c). Algorithm to compute a generating-set for $H^\perp$. We follow the lines of an algorithm shown in [40]. By definition, an element $g$ belongs to $H^\perp$ if and only if $\chi_g(h) = 1$ for all $h \in H$ and, by linearity, if and only if $\chi_g(h_j) = 1$ for each generator of $H$. Using equation 1.15 this condition is equivalent to

$$\exp \left\{ 2\pi i \left( \frac{g_1 h_{j_1}}{N_1} + \ldots + \frac{g_d h_{j_d}}{N_d} \right) \right\} = 1 \quad (5.6)$$

These conditions can be turned into a system of modular equations. Substitute $g$ with an unknown element $X$. Compute $M := \lcm(N_1, \ldots, N_d)$ using Euclid’s algorithm and the numbers $\alpha_i := M/N_i$, then $X$ is an element of the orthogonal if and only if it is a solution of the following equivalent system of equations.

$$\alpha_1 h_{11} X_1 + \alpha_2 h_{12} X_2 + \ldots + \alpha_d h_{1d} X_d = 0 \quad (\text{mod } M) \quad (5.7)$$

$$\alpha_1 h_{21} X_1 + \alpha_2 h_{22} X_2 + \ldots + \alpha_d h_{2d} X_d = 0 \quad (\text{mod } M) \quad (5.8)$$

$$\vdots$$

$$\alpha_1 h_{r1} X_1 + \alpha_2 h_{r2} X_2 + \ldots + \alpha_d h_{rd} X_d = 0 \quad (\text{mod } M) \quad (5.9)$$

As it is proven in [40], if we sample $t + \lceil \log |G| \rceil$ random solutions of this system of equations we will obtain a generating set of $H^\perp$ with probability $p$ exponentially close to one: $p \geq 1 - 1/2^t$, thus, if
we find a polynomial number of random solutions it suffices for our purposes. In order to sample the solutions of the system of equations re-write it as \( AX = 0 \pmod{M} \), where \( A \) is a rectangular matrix over the integers modulo \( M \). Using an algorithm given in [59] it is possible to compute the Smith normal decomposition of \( A \), i.e., to obtain a diagonal matrix \( D \) and two invertible matrices \( U, V \) such that \( D = UDV \). Substituting, the system of equations can be written as \( DY = 0 \pmod{M} \) with \( X = VY \). Now it is possible to randomly compute solutions of \( DY = 0 \pmod{M} \) using Euclid’s algorithm since this is a system of equations of the form \( d_iy_i = 0 \pmod{M} \). Finally, computing \( X = VY \) one obtains a random element of the orthogonal group \( H^\perp \) as desired. We repeat the procedure until we have a poly-size generating-set of \( H^\perp \).

(b) Decide membership. Since we already know how to compute a generating-set of \( H^\perp \), it is easy to check if a given element \( g \) belongs to \( H \). First compute a generating-set \( \langle g_1, \ldots, g_r \rangle \) of \( H^\perp \). Then, by definition, \( g \) belongs to \( H \) if and only if \( \chi_g(g_i) = 1 \) for all generators of \( H^\perp \). Since there are a polylog(\( |G| \)) number of them and this can be done efficiently using modular arithmetic (use the same equations we used to prove (a)) we are done.

5.3 Algorithms for semi-direct products

It is clear now that for any \( \mathbb{Z}_p \rtimes \varphi A \) we can compute quantities like \( \varphi(a) \). However, the ability of computing powers of \( M_\varphi \) does not necessarily come for granted. We discuss how this relates to performing group-multiplication and exponentiation.

Definition 5.3.1. If there exists a classical algorithm to multiply elements in \( \mathbb{Z}_p \rtimes \varphi A \) we say that group satisfies the condition \( \text{MULT} \). Similarly, if there exists a classical algorithm to exponentiate elements in \( \mathbb{Z}_p \rtimes \varphi A \) we say that group satisfies \( \text{EXP} \).

Lemma 5.3.2. Given any semidirect group of the form \( \mathbb{Z}_p \rtimes \varphi A \):

1. \( \mathbb{Z}_p \rtimes \varphi A \) fulfils \( \text{MULT} \) iff the quantities \( \varphi^n(a) \) can be computed efficiently.
2. \( \mathbb{Z}_p \rtimes \varphi A \) fulfils \( \text{EXP} \) iff the quantities \( \Phi_m^n(a) \) can be computed efficiently.
3. \( \text{EXP} \) implies \( \text{MULT} \).
4. If \( p = O(\text{polylog}|A|) \) both \( \text{MULT} \) and \( \text{EXP} \) are always fulfilled.

Proof: The first proposition comes from \( (m,a)(n,b) = (m,a) + (n,\varphi^n(b)) \). The second from \( (m,a)^n = (mn,\Phi_m^n(a)) \). The third from \( \Phi_m^n(a) - \Phi_m^1(a) = \varphi^m(a) \). We prove 4 now. First, Lemma 5.2.1 shows that we need time \( O(p \times \text{polylog}|A|) \) to compute \( \varphi^n(a) = M^n_p \cdot a \). Second, consider this recursive procedure:

- Init: \( \Phi_m^0(a) := 0 \).
- From 0 to \( n \) repeat \( \Phi_m^{i+1}(a) := a + \varphi^m(\Phi_m^i(a)) \).

This algorithm uses \( O(p^2 \times \text{polylog}|A|) \) time to compute \( \Phi_m^n(a) \).

For most practical applications a group \( \mathbb{Z}_p \rtimes \varphi A \) should satisfy \( \text{EXP} \) and hence also \( \text{MULT} \). In fact, to the best knowledge of the authors, all semidirect products of the form \( \mathbb{Z}_p \rtimes \varphi A \) considered in the Hidden Subgroup Literature satisfy \( \text{EXP} \): Some examples are: dihedral and \( p \)-hedral groups \( \mathbb{Z}_p \rtimes \mathbb{Z}_N \); wreath products \( \mathbb{Z}_p \rtimes \mathbb{Z}_p^p \); Heisenberg groups \( \mathbb{Z}_p \rtimes \mathbb{Z}_p^r \) for fixed \( r \). A trivial instance are abelian groups \( \mathbb{Z}_p \rtimes A \), whose classical algorithms to add, multiply and exponentiate are folklore.

Convention. Throughout this work we will impose that our groups must fulfil \( \text{EXP} \). Otherwise the groups would not be very interesting for applications, as the following example illustrates: to compute elements of a cyclic subgroup given its generator \( \langle (m,a) \rangle \) one needs to perform exponentiations. As a consequence, without \( \text{EXP} \) we can not describe subgroups with poly-size generating sets.
5.3. ALGORITHMS FOR SEMI-DIRECT PRODUCTS

5.3.1 Generating-sets

As an example of this, consider the following application: we prove now that, given any poly-size set of generators for a subgroup of type $2$ in lemma 4.3.2, it is possible to find a new poly-size set of generators of the subgroup with the canonical form $H = \langle (1, d), H_A \rangle$ if and only if there exists an efficient classical algorithm to exponentiate inside $\mathbb{Z}_p \rtimes \varphi A$.

**Lemma 5.3.3.** Given a group $\mathbb{Z}_p \ltimes A$ in which we can efficiently exponentiate with a classical algorithm. Given a subgroup $H$ of $\mathbb{Z}_p \ltimes A$ not contained in $A$ and described by a poly-size set of generators $\langle h_1, \ldots, h_r \rangle$. Then, there exists an efficient classical algorithm to find a new poly-size set of generators of the canonical form $H = \langle (1, d), h'_1, \ldots, h'_s \rangle$ and $H_A = \langle h'_1, \ldots, h'_s \rangle$.

**Proof:** Assume, w.l.o.g, that $h_r = (m, a)$ with $m \neq 0$. Since $\mathbb{Z}_p$ is a field, there exists a multiplicative inverse $m^{-1}$ of the number $m$ in $\mathbb{Z}_p$ and it can be found efficiently using Euclid’s algorithm. We use the exponentiation algorithm to compute $(m, a)^{m^{-1}} = (mm^{-1}, \Phi_{m}^{m^{-1}}(a)) \equiv (1, d)$. Now realise that any generator $h_i = (m_i, a_i)$ fulfils

\[(m_i, a_i) = (0, a_i - \Phi_{m_i}^{m_i}(d))(1, d)^{m_i}\]  (5.11)

Then, for any $h_i = (m_i, a_i)$ include the element $(0, b_i) := (0, a_i - \Phi_{m_i}^{m_i}(d))$ in a new set $S_0$. We have constructed a simpler generating set:

$H = \langle (1, d), S_0 \rangle = \langle (1, d), (0, b_1), \ldots, (0, b_r) \rangle$  (5.12)

This is almost what we wanted, since there might be elements in $H_A$ that are combinations of elements from $S_0$ and powers of $(1, d)$. To include them, construct $S_1 := S_0 \cup \varphi(S_0)$ which is invariant under the automorphism $\varphi$ and still contained in $H_A$ which is equal to $\varphi(H_A)$ because of lemma 4.3.2. The set defines a new poly-size generating set $H = \langle (1, d), S_1 \rangle$ with the following property

\[\langle (1, d) \rangle \langle S_1 \rangle = \langle S_1 \rangle \langle (1, d) \rangle\]  (5.13)

which holds because $(1, d)S_1(1, d)^{-1} = \varphi(S_1) = S_1$. This property guarantees that any possible combination of generators (including their inverses) can be re-ordered in the form $(1, d)^m(0, h)$ with $(0, h) \in \langle S_1 \rangle$. As a consequence, the only elements in $H_A$ that could not be in $\langle S_1 \rangle$ are those of the form $\langle (1, d) \rangle \cap A = \langle (0, \Phi_{m}^{m}(d)) \rangle$. Hence if we take the set of generators $\langle (1, d), S_A \rangle$ with $S_A := S_1 \cup \{(0, \Phi_{m}^{m}(d))\}$ it has the desired form. $\blacksquare$
Part III

Simulation of semi-direct quantum Fourier transforms
Chapter 6

Efficient quantum Fourier transforms for semi-direct product groups

Summary

We find new quantum circuits to implement quantum Fourier transforms for a large class of non-abelian groups, the semi-direct products $\mathbb{Z}_p \ltimes \varphi A$. We give a general formula that works for any group and allows to use different choices bases for high dimensional irreducible representations. In our design we use the abelian quantum Fourier transform $F_A$ of $A$ as a subroutine. Our circuits are always efficient whenever the prime number $p$ scales as $O(\text{polylog}|A|)$, and, in the most general case, we show that the efficiency depends on certain structural properties of the automorphism $\varphi$ used in the definition of the semi-direct group. In both regimes we find efficient QFTs for non-abelian groups which are studied in Quantum Computation [16, 17, 18, 19, 20].

The structure of the chapter is as follows. In section 6.1 we summarise the representation theory needed to understand the mathematical formulas of this chapter. In section 6.2 we introduce a quantum circuit which performs a canonical unitary Fourier transformation (eq. 6.18) for any semidirect product $\mathbb{Z}_p \ltimes \varphi A$. We explain in detail the main features of this circuit and discuss its efficiency. In section 6.3 we show how to use our transforms to perform Weak and Strong Fourier Sampling. In section 6.4 we show how to implement QFTs for different choices of basis for the irreducible representations of $\mathbb{Z}_p \ltimes \varphi A$, given that the basis itself can be efficiently implemented as well. In section 6.5 we comment how this circuit relates to previous works and give interesting examples of semi-direct products for which our circuits are efficient.

In appendix A we explain some advance tools from representation-theory which we have used to find a complete-set of irreducible representations of $\mathbb{Z}_p \ltimes \varphi A$ such as induced representations and the little group method of Wigner and Mackey. Our presentation is oriented to quantum physicists and uses the bra-ket Dirac notation. In appendix D we survey the original ideas we used to develop our quantum circuits.

Conventions

Throughout this chapter, the regular representation gates of the abelian direct group $\mathbb{Z}_p \times A$ are denoted $X(m, a) := X_p(m) \otimes X_A(b)$ and the diagonalised regular representation gates as $Z(m, a) := Z_p(m) \otimes Z_A(b)$ and. These gates correspond to well-known generalised Pauli matrices. For the regular representations of the non-abelian semidirect product $\mathbb{Z}_p \ltimes \varphi A$ we use bold letters and the usual labels $L, R$ to denote the left- $X_L$ and right- $X_R$ regular representations and their block-diagonalised versions $Z_L, Z_R$. We will use the greek letter sigma $\sigma$ to denote the high-dimensional irreducible representations of our groups which must not be confused with the Pauli matrices.
6.1 Representation theory of semi-direct products

In this section we give explicit formulas for the irreducible-representation and the regular-representation unitary gates of a semi-direct product $\mathbb{Z}_p \ltimes \varphi A$. We limit ourselves to present these matrices and explain their main features. The technical methods we have used to obtain expressions for the irreducibles are given in appendix A.

6.1.1 The $\varphi$-partition of the abelian group $A$

A powerful result from representation theory (cf. section 6.1.3) shows that the representation theory of $\mathbb{Z}_p \ltimes \varphi A$ is strongly related to the properties of group action from $\mathbb{Z}_p$ on the dual group $\hat{A}$, which we will introduce now. Any action form a group $G$ on a set $\hat{A}$ is completely specified by giving the bijective functions that the generators of the group induce on the set. In our case, the group is generated by one single element which defines a function $\hat{\varphi} : \hat{A} \to \hat{A}$.

**Proposition 6.1.1.** For any group semidirect group $\mathbb{Z}_p \ltimes \varphi A$ the function $\hat{\varphi} : \hat{A} \to \hat{A}$ defined as

$$\chi_{\hat{\varphi}(a)} := \chi_a \circ \varphi^{-1}$$

(6.1)

is a group automorphism of $\hat{A}$ of order $p$ which defines a group action from $\mathbb{Z}_p$ to $\hat{A}$ via the correspondence $m \cdot \chi_a = \chi_{\hat{\varphi}^m(a)}$.

**Proof:** See the proof of proposition A.1.2.

There are some immediate consequences of this property. First, because $\hat{A}$ and $A$ are isomorphic for any abelian group, $\hat{\varphi}$ is also an automorphism of $A$. Second the groups $\mathbb{Z}_p = \langle \hat{\varphi} \rangle$ are isomorphic via $m \leftrightarrow \hat{\varphi}^m$. Last and crucial to us, it is known from group theory that any group action on a set defines a partition into different disjoint subsets that are invariant under the action and are known as orbits of the partition. Therefore, $\hat{\varphi}$ induces a partition into $A$ defined by a set of representatives $R$ that we will call the ‘$\varphi$-partition’ of $A$.

The orbits of the $\varphi$-partition can be easily classified. Use that the stabiliser subgroup $H_a$ of a given element $a \in A$ is a subgroup of $\langle \varphi \rangle \simeq \mathbb{Z}_p$: since the automorphism $\varphi$ has prime order $p$ and the order of any subgroup must divide the order of the total group, the stabiliser subgroup $H_a$ must have either 0 or $p$ elements. As the number of elements of the stabiliser group $H_a$ is related to the number of the elements of the orbit $O_a$ through the formula $|H_a| = p/|O_a|$, it follows that the orbits of the partition must have also 1 or $p$ elements. These properties are summarised in box 6.1.

The action of the automorphism $\varphi$ on $\hat{A}$ of equation (6.1) induces (via the isomorphism $\chi_a \leftrightarrow a$) a partition of $A$ into disjoint orbits of two different types:

- Orbits with one element $O_{a_r} = \{a_r\}$ represented by $a_r = \varphi(a_r)$.
- Orbits with $p$-elements $O_{b_r} = \{b_r, \varphi(b_r), \ldots, \varphi^{p-1}(b_r)\}$ represented by $b_r \neq \varphi(b_r)$.

Denoting by $\hat{R}_1$ and $\hat{R}_p$ the sets of representatives for each kind or orbit, the partition of $A$ can be written as

$$A = \bigcup_{a_r \in \hat{R}_1} O_{a_r} \bigcup_{b_r \in \hat{R}_p} O_{b_r}$$

(6.2)

where all unions are disjoint.

**Box 6.1:** The $\varphi$-partition of the group $A$

6.1.2 The dual automorphism

The automorphisms $\varphi$ and $\varphi$ are actually dual functions, meaning that they are Fourier-transformed copies of each other. To prove this we use the unitary gates that implement the automorphisms and the abelian quantum Fourier transform $F_A$ defined in eq. 1.19.
6.1. REPRESENTATION THEORY OF SEMI-DIRECT PRODUCTS

Proposition 6.1.2. Consider the unitary gates $U_\varphi$ and $U_{\hat{\varphi}}$ implementing the automorphisms $\varphi$ and $\hat{\varphi}$. These gates are dual to each other, meaning that they are related by a change of basis which is the quantum Fourier transform of $A$.

$$U_\varphi := \sum_a |\varphi(a)\rangle \langle a| = F_A^\dagger U_{\hat{\varphi}} F_A$$ (6.3)

Proof:

$$F_A U_\varphi F_A^\dagger = F_A \left( \sum_b \chi^*_a(b) |\varphi(b)\rangle \langle a| \right) F_A^\dagger$$ (6.4)

$$= F_A \left( \sum_{a,b} \chi^*_a(b) |\varphi(b)\rangle \langle a| \right) \left( \sum_{a',b'} \chi_{\hat{\varphi}}^*(a') |b'\rangle \langle a'| \right)$$ (6.5)

$$= \sum_{a,a'} \left( \sum_{b' \in G} \chi_{\hat{\varphi}}^*(b') \chi^*_a(a') \right) |a'\rangle \langle a| = \sum_{a,a'} \delta_{a',\hat{\varphi}(a)} |a'\rangle \langle a|$$ (6.6)

$$= \sum_a |\hat{\varphi}(a)\rangle \langle a| = U_{\hat{\varphi}}$$ (6.7)

Where we used the orthogonality of character functions (1.16).

6.1.3 Irreducible representations

Now we present a full-characterisation of the irreducible representations of $\mathbb{Z}_p \ltimes_\varphi A$ which exploits a powerful one-to-one correspondence\[1\] between the orbits of the $\hat{\varphi}$-partition of $\mathbb{Z}_p \ltimes_\varphi A$ and its irreducible representations. The classification is presented in box 6.2, which summarises in simple words the content of a theorem A.2.1 that we prove in appendix A.

We also give in appendix A further details about the main technical tool we have used to characterise the irreps, which is known as Wigner-Mackey’s little group method. The method involves a few relatively basic representation-theoretical techniques which we present using common notation in quantum physicist. The original reference we have followed is Serre’s textbook [60].

6.1.4 Regular representations

We end this section giving unitary expressions for the left and right regular representations of $\mathbb{Z}_p \ltimes_\varphi A$. The left regular representation can be obtained straightforwardly from the multiplication rule 4.1 and the definition 1.6

$$X_L(n, b) = X(n, b) (I_p \otimes U_{\varphi})^n$$ (6.11)

For the right regular representation is useful to use that it is an homomorphism and thus it can be split in to terms:

$$X_R(m, a) = X_R(0, a) X_R(1, 0)^m$$ (6.12)

And then use again the equation 4.1 and definition 1.6 to obtain

$$X_R(1, 0) := X_p^\dagger \otimes I_A$$ (6.13)

$$X_R(0, a) := \sum_n |n\rangle \langle n| \otimes X_A(-\varphi^n(a))$$ (6.14)

\[1\] Note: this “correspondence” is an extension rule defined via induction as it can be consulted in appendix A.
CHAPTER 6. EFFICIENT QFTS FOR SEMI-DIRECT PRODUCT GROUPS

For any a semi-direct product of the form $\mathbb{Z}_p \ltimes \varphi A$, the irreducible representations of the group are either 1 or $p$ dimensional. Moreover, there is a canonical way to obtain a complete set of inequivalent irreducible representations of the group from the $\hat{\varphi}$-partition of $A$. Choosing the a set of representatives $\hat{R} := \hat{R}_1 \cup \hat{R}_p$ as in box 6.1 all irreps of $\mathbb{Z}_p \ltimes \varphi A$ belong to one of the following classes:

1. **One-dimensional irreps**. Any element $a_r$ of the group $A$ fulfilling $a_r = \hat{\varphi}(a_r)$, representative of an orbit with one element $O_{a_r} = \{a_r\}$, defines $p$ one-dimensional irreducible representations $\chi_{m,a_r}$

$$\chi_{m,a_r}(n,c) := \omega_p^{mn} \chi_{a_r}(c) \quad (6.8)$$

where $m$ takes all values from 0 to $p - 1$, $\omega_p$ is the primitive $p$th root of the unity and $\chi_{a_r}$ is an abelian character function. Two irreducible representations $\chi_{m,a_r}, \chi_{m',a'_r}$ are equivalent if and only if $(m, a_r) = (m', a'_r)$ and any one-dimensional irreducible representation of the group has to be of this form.

2. **High-dimensional irreps**. For any other representative element of the group $b_r \in \hat{R}_p$, which fulfills the opposite condition $b_r \neq \hat{\varphi}(b_r)$, the orbit of $p$-elements represented by the element $O_{b_r} = \{b_r, \hat{\varphi}(b_r), \ldots, \hat{\varphi}^{p-1}(b_r)\}$ defines one $p$-dimensional irrep $\sigma_{b_r}$.

$$\sigma_{b_r}(n,c) := Z_{O_{b_r}}(c) X_p(n) \quad (6.9)$$

where $X_p(n)$ and $Z_{O_{b_r}}(c)$ are defined as

$$X_p(n) := \sum_{i=0}^{p-1} |i+n\rangle \langle i| \quad Z_{O_{b_r}}(c) := \sum_{i=0}^{p-1} \chi^{\hat{\varphi}(b_r)}(c) |i\rangle \langle i| \quad (6.10)$$

and two irreducible representations $\sigma_{b_r} \simeq \sigma_{b'_r}$ are equivalent if and only if $b_r$ and $b'_r$ belong to the same orbit, which guarantees that the definition $\sigma_{b_r}$ is independent of the of the choice of representative.

Notice that although the correspondence between orbits and irreducible representations does not depend on the choice or representatives of the partition, a particular choice must be taken when one implements this gates as a quantum circuit. For rigorous proofs of the claims of this box consult section A and theorem A.2.1.

Box 6.2: Classification of the irreducible representations of $\mathbb{Z}_p \ltimes \varphi A$

### 6.2 An efficient canonical circuit for the semi-direct QFT

In this section we show how a particular quantum circuit $\mathcal{F}$ that implements a quantum Fourier transform for the groups $\mathbb{Z}_p \ltimes \varphi A$ and can be written in canonical form $[1.12]$. From now, we will restrict to a class of subgroups groups inside the family $\mathbb{Z}_p \ltimes \varphi A$ for which our circuit is always efficient. The quantum Fourier transform $\mathcal{F}$ will be used in next sections as a basic subroutine to implement more general Fourier transforms.

#### 6.2.1 Conventions

From now on we will no-longer consider arbitrary semi-direct product groups of the form $\mathbb{Z}_p \ltimes \varphi A$ but only those for which certain arithmetical operations can be performed efficiently on a classical computer. Strictly speaking, we define a new condition ARITH which all the groups we consider must fulfill.

**Definition 6.2.1.** We say that a semi-direct group $\mathbb{Z}_p \ltimes \varphi A$ satisfies the condition ARITH if and only there are known efficient classical algorithms to perform the operations listed below

1. **Multiplication and exponentation.** As we saw in section 5.3 this implies that the functions $\varphi^m$ and $\Phi_m$ can be efficiently computed.
6.2. AN EFFICIENT CANONICAL CIRCUIT FOR THE SEMI-DIRECT QFT

2. Computing dual automorphism functions $\hat{\varphi}^m$.

3. Solving the following problem. Given a system of representatives $\hat{R}$ of the $\hat{\varphi}$-partition of $A$ and an element $x \in A$, find the representative $b_r$ such that $x \in O_{b_r}$ and return the number $i$ such that $x = \hat{\varphi}^i(b_r)$.

The first condition in ARITH guarantees the efficiency of all the algorithms for semi-direct groups seen in section 5.3. The second and the third conditions help to ‘study the structure’ of $\hat{\varphi}$-partition of the abelian group $A$. Condition 3, for example, guarantees that:

1. There is a classical routine to decide whether two irreducible representations of box 6.2 are equivalent.

2. Given two equivalent $p$-dimensional irreducibles $\sigma_{b_r} \simeq \sigma_{b'_r}$ one can compute the change of basis that relates them (cf. box 6.2).

The class of semi-direct products $\mathbb{Z}_p \rtimes \phi A$ satisfying the condition ARITH is also big enough to include interesting examples of groups studied in quantum computation such as: the Dihedral Group $\mathbb{Z}_2 \rtimes \mathbb{Z}_N$ [14], wreath products $\mathbb{Z}_2 \rtimes \mathbb{Z}_2^n$ [17], groups $\mathbb{Z}_p \rtimes \mathbb{Z}_p^r$ for fixed $r$ [43] (including the Heisenberg Group) and other unnamed semi-direct products such as $\mathbb{Z}_2 \rtimes \mathbb{Z}_n$ [19] and $\mathbb{Z}_p \rtimes \mathbb{Z}_m$ [20].

6.2.2 The quantum circuit $\mathcal{F}$

We will now describe the aforementioned quantum circuit $\mathcal{F}$ and discuss its efficiency. Our circuits make use of several already-defined unitary gates as sub-routines, namely: the Fourier transform $F_A$ of the abelian group $A$, the sign-flip automorphism gate $U_{(\cdot)}$, and the Pauli gate $X_p(i)$. For remembrance:

$$F_A := \frac{1}{\sqrt{|A|}} \sum_{a,b \in A} \chi_a(b)|a\rangle\langle b| \quad U_{(\cdot)} := \sum_{a \in A} |a\rangle\langle -a| \quad X_p(i) = \sum_{n=0}^{p-1} |n+i\rangle\langle n|$$

Also, we use a set of representatives $\hat{R}$ of the $\hat{\varphi}$-partition of $A$ as in box 6.2.

**Theorem 6.2.2.** For any group $\mathbb{Z}_p \rtimes \varphi A$, the quantum circuit $\mathcal{F} := CU_{\hat{R}}CF_A(1 \otimes F_A)$ shown in figure 6.1 implements a canonical quantum Fourier transform. The first gate $F_A$ is the quantum Fourier transform

$$|m\rangle \quad F_p \quad U_{\hat{R}} \quad |a\rangle$$

![Figure 6.1: $\mathcal{F}$ quantum Fourier transform of $\mathbb{Z}_p \rtimes \varphi A$](image)

of the abelian group $A$ and the controlled-unitaries are defined as follows:

1. The first gate $CF_p$ Fourier-transforms the first register if the second register contains an element fulfilling $a = \hat{\varphi}(a)$ and does nothing in the opposite case $b \neq \hat{\varphi}(b)$

$$CF_p := \sum_{m,a=\hat{\varphi}(a)} F_p \otimes I_A |m,a\rangle\langle m,a| + \sum_{n,b \neq \hat{\varphi}(b)} |n,b\rangle\langle n,b|$$  (6.15)

2. The second gate $CU_{\hat{R}}$ performs a change of basis of the high-dimensional irreps acting on the first register controlled on the second and using $\hat{\varphi}$-partition of $A$. If the second register contains an invariant representative $a_r = \hat{\varphi}(a_r)$ it does nothing. Otherwise, it contains a non-invariant
element which can be expressed as \( b = \hat{\varphi}(b_r) \) for some representative \( b_r \in \hat{R} \) and the gate applies a change of basis \( U_i \) controlled on the number \( 'i' \).

\[
CU_{\hat{R}} := \sum_{m,a_r} |m,a_r\rangle \langle m,a_r| + \sum_{m,\hat{\varphi}(b_r)} U_i \otimes I_A |n,\hat{\varphi}(b_r)\rangle \langle n,\hat{\varphi}(b_r)|
\]

(6.16)

where the gate \( U_i \) is a \( p \times p \) unitary defined as

\[
U_i := X_p(i)U = \sum_{n=0}^{p-1} |i - n\rangle \langle n|
\]

(6.17)

Moreover, if the group \( \mathbb{Z}_p \ltimes \varphi A \) fulfils ARITH the circuit \( \mathcal{F} \) can be efficiently implemented.

**Proof:** For clarity, we dedicate the next section to prove that the circuit \( \mathcal{F} \) is a canonical QFT separately; here we just prove that the circuit \( \mathcal{F} \) is efficient. First, the gates \( F_A \) and \( CF_p \) can be implemented efficiently because any abelian-QFT can be implemented efficiently (cf. section 1.5) and any “controlled”-\( U \) gate can be implemented efficiently if its “uncontrolled” version \( U \) can be implemented efficiently [9]. Moreover, the condition ARITH implies that \( CU_{\hat{R}} \) can be computed efficiently with a classical circuit and, hence, also with a quantum circuit.

**6.2.3 The unitary gate \( \mathcal{F} \) implements a canonical quantum Fourier transform**

In this section we prove that the quantum circuit \( \mathcal{F} \) implements a canonical Fourier transform. To do so, we recall the expression of the canonical non-abelian QFT from chapter 4

\[
F_G := \sum_{g \in G} \sum_{\sigma \in G} \sqrt{\frac{d_\sigma}{|G|}} \sum_{i,j} [\sigma(g)]_{j,i} |i,j,\sigma\rangle \langle g|
\]

(6.18)

Equation (6.18) can be read as follows. The unitary \( F_G \) sends any input state \( |g\rangle \) to a non-trivial and non-uniform superposition over the whole computational-basis of the Hilbert space. The superposition is created by re-labelling the computational-basis states giving them new unique names \( |i,j,\sigma\rangle \); then, these new names indicate how much amplitude must be given to each state (a quantity \( \sqrt{d_\sigma/|G|} [\sigma(g)]_{j,i} \) if the input state was labelled by the group element \( g \).

Equation (6.18) does not suggests a particular re-labelling \( |i,j,\sigma\rangle \) of the computational-basis and it does not tell us whether a good-choice that simplifies the implementation of the transform exists. For our semi-direct QFT we have used a “natural” scheme based on the correspondence between the irreducible representations and the orbits of the \( \hat{\varphi} \)-partition of the group shown in box 6.2. The scheme is summarised in box 6.3.

With the re-labelling scheme proposed in box 6.3 we prove the following lemma that gives itself a proof of theorem 6.2.2.

**Lemma 6.2.3.** The quantum circuit \( \mathcal{F} \) from theorem 6.2.2 can be written in the canonical form 6.18 and, therefore, it is a quantum Fourier transform.

**Proof:** Remind the expression of the chosen basis \( U_i := X_p(i)U_{(-)} = \sum_{n=0}^{p-1} |i - n\rangle \langle n| \). Multiply \( CF_p \) and \( CU_{\hat{R}} \) together, obtaining a block-diagonal[4] matrix \( B_D := CU_{\hat{R}}CF_p \). Use the re-labelling from box 6.3 we have just explained to transform the computational basis states as \( |\hat{\varphi}(b_r)\rangle \equiv |i,b_r\rangle \) to obtain:

\[
B_D := \frac{1}{\sqrt{p}} \sum_{m,n,a_r} \omega_p^{mn} |m,a_r\rangle \langle n,a_r| + \sum_{n,i,b_r} |i - n\rangle \langle n| \otimes |i,b_r\rangle \langle i,b_r| 
\]

(6.19)

[4]It acts independently on the blocks of the decomposition (1.4.1). This can be seen better in box 6.4.
6.2. **AN EFFICIENT CANONICAL CIRCUIT FOR THE SEMI-DIRECT QFT**

For the semi-direct group \( \mathbb{Z}_p \ltimes \varphi A \) there exist a natural re-labelling of the computational-basis states \(|m, a\rangle\) that we use in this work to design quantum measurements based on non-abelian Fourier sampling, described by the following scheme.

- The \( p \) computational-basis states \(|m, a_r\rangle\) labelled by an invariant representatives \( a_r = \varphi(a_r) \) and \( m \) from \( 0 \) to \( p - 1 \) correspond in a one-by-one way to \( p \)-inequivalent one dimensional irreducible representations \( \chi_{m,a_r} \). There is no need to re-label this elements.

- The computational basis states \(|m, \varphi^i(b_r)\rangle\) labelled by the elements of the orbit represented by \( b_r \neq \varphi(b_r) \), and for any \( m \) from \( 0 \) to \( p - 1 \), correspond in a one-to-one way to the matrix coefficients of the irreducible representation \( \sigma_{b_r} \). In this case, we can re-label the states in a simple way as \(|m, \varphi^i(b_r)\rangle \equiv |m, i, b_r\rangle\). Notice that in this case \( m, i \in \mathbb{Z}_p \), making a total of \( p^2 \) states for each \( b_r \).

From the classification of the irreps provided in box 6.2 it is immediate that the new states are of the form \(|i, j, \sigma\rangle\) and that there are \(|G|\) of them, hence, the re-labelling is well-defined.

**Box 6.3**: The re-labelling \(|i, j, \sigma\rangle\)

Now we compute \( \mathcal{F} = B_D(I_p \otimes F_A)\):

\[
\mathcal{F} = B_D(I_p \otimes F_A) \\
= B_D \left( \frac{1}{\sqrt{|A|}} \sum_{n, a_r, c} \chi_{a_r}(c) |n, a_r\rangle \langle n, c| + \sum_{n, b_r, c} \chi_{\varphi^i(b_r)}(c) |n, i, b_r\rangle \langle n, c| \right) \\
= \frac{1}{\sqrt{|p|A|}} \left( \sum_{m, m, a_r, c} \omega_m^{n,m} \chi_{a_r}(c) |m, a_r\rangle \langle n, c| + \sqrt{p} \sum_{n, i, b_r, c} \chi_{\varphi^i(b_r)}(c) |i - n, i, b_r\rangle \langle n, c| \right) \\
= \frac{1}{\sqrt{|G|}} \left( \sum_{m, n, a_r, c} \chi_{m,a_r}(n, c) |m, a_r\rangle \langle n, c| + \sqrt{p} \sum_{m, n, i, b_r, c} \delta_{m+n} \chi_{\varphi^i(b_r)}(c) |m, i, b_r\rangle \langle n, c| \right) \\
= \sum_{(n, c) \in G} \sum_{\sigma \in \tilde{G}} \sqrt{d_{\sigma}} \sum_{i,j=0}^{d_{\sigma}} |\sigma(n, c)\rangle |i, j, \sigma\rangle \langle n, c| \\
= \sum_{(n, c) \in G} \sum_{\sigma \in \tilde{G}} \sqrt{d_{\sigma}} |\sigma(n, c)\rangle |i, j, \sigma\rangle \langle n, c| (6.21)
\]

In the last equality, we have used the explicit formulas of the irreducible representations of \( G = \mathbb{Z}_p \ltimes \varphi A \) given in box 6.2. The final equation (6.21) is the canonical quantum Fourier Transform of any finite group \( \mathbb{Z}_p \ltimes \varphi A \).

**6.2.4 Block-diagonal formulas**

Finally, we end this section showing that the quantum Fourier transform \( \mathcal{F} \) given in theorem 6.2.2 block-diagonalises the left and right regular representations ‘exactly’ as we said in the original definition 1.4.1

\[
Z_L(n, c) := \mathcal{F} X_L(n, c) \mathcal{F}^\dagger = \bigoplus_{\sigma \in \tilde{G}} I_{d_{\sigma}} \otimes \sigma(n, c) \\
Z_R(n, c) := \mathcal{F} X_R(n, c) \mathcal{F}^\dagger = \bigoplus_{\sigma \in \tilde{G}} \sigma(n, c)^* \otimes I_{d_{\sigma}} \\
(6.22) (6.23)
\]

By ‘exactly’ we mean in the decompositions of \( Z_L \) and \( Z_R \) the irreducibles representations are expressed in ‘exactly’ the same basis, which is the one obtained from the little group method and it is defined in box 6.2. We prove this formulas in the appendix B but we include a summary here, box 6.3.
The following equations described the unitary gates $\mathbf{Z}_L$, $\mathbf{Z}_R$ obtained applying the quantum Fourier transform $\mathcal{F}$ to the regular representations $X_L$, $X_R$ as a change of basis.

$$\
\mathbf{Z}_L(n, c) = \sum_{m,a_r} \chi_{m,a_r}(n,c) |m,a_r\rangle\langle m,a_r| + \sum_{b_r} I_p \otimes \sigma_{b_r}(n,c) \otimes |b_r\rangle\langle b_r| \quad (6.24) \\
= \bigoplus_{m,a_r} \chi_{m,a_r}(n,c) \bigoplus_{b_r} I_p \otimes \sigma_{b_r}(n,c) \quad (6.25) \\
\mathbf{Z}_R(n, c) = \sum_{m,a_r} \chi^*_{m,a_r}(n,c) |m,a_r\rangle\langle m,a_r| + \sum_{a,b_r} \sigma_{b_r}^*(n,c) \otimes I_p \otimes |b_r\rangle\langle b_r| \\
= \bigoplus_{m,a_r} \chi^*_{m,a_r}(n,c) \bigoplus_{b_r} \sigma_{b_r}^*(n,c) \otimes I_p \quad (6.26) \\
\mathbf{Z}_R(n, c)^{CV} = \bigoplus_{m,a_r} \chi^*_{m,a_r}(n,c) \bigoplus_{b_r} \left[ V \sigma_{b_r}^*(n,c)V^\dagger \right] \otimes I_p \quad (6.28)
$$

The above equations are written using the re-labelling $|i,j,\sigma\rangle$ of box 6.3. From them equations it is easy to realise how to implement other quantum Fourier transforms in other different basis. For simplicity, we show how to design a unitary gate $CV$ to do this for $\mathbf{Z}_R$. The unitary $CV$ first reads the label of the irrep and checks whether it corresponds to a one dimensional irrep $|i,j,\sigma\rangle \equiv |m,a_r\rangle$ or to a high dimensional irrep $|i,j,\sigma\rangle \equiv |m,i,b_r\rangle$. In the former case the state is left unchanged, in the latter the transform applies a change of basis $V \otimes I_A$ over the state. The final gate after the change of basis is:

$$\mathbf{Z}_R(n, c)^{CV} = \bigoplus_{m,a_r} \chi^*_{m,a_r}(n,c) \bigoplus_{b_r} \left[ V \sigma_{b_r}^*(n,c)V^\dagger \right] \otimes I_p$$

The transform $CV$ can be easily implemented using an one-qubit ancillary system.

Box 6.4: Fourier transformed regular-representation gates

6.3 Weak and Strong Fourier Sampling

In this section we show how to use our quantum Fourier transform 6.2.2 to perform weak and strong Fourier sampling. From the original definitions (section 2.3.3) it is immediate that we can implement these sampling schemes are quantum projective measurements on the re-labelled computational basis we introduced in box 6.3. In the particular case of groups of the form $\mathbb{Z}_p \ltimes_{\varphi} A$ the $\hat{\varphi}$-partition of the group $A$ (cf. box 6.1) can be used to implement both measurements in a simple fashion, which we proceed to explain.

6.3.1 Weak Fourier sampling

The following measuring scheme allows to implements weak Fourier sampling on any initial coset-state $|(m,a)H\rangle$ of the semi-direct group $\mathbb{Z}_p \ltimes_{\varphi} A$. Remember that, the objective of this measurement is to measure the label of an irreducible representation. In our case, the labels are the representatives of the orbits $a_r$, $b_r$ described in box 6.2.

1. Apply the transform $\mathcal{F}$ on $|(m,a)H\rangle$.

2. Measure the abelian register in the original computational-basis $|m,a\rangle$. This is a quantum projective measurement defined by projectors $P_a := I_p \otimes |a\rangle\langle a|$ for any $a \in A$. The output of this measurement is an element $x \in A$.

3. Check whether $x$ satisfies $x = \hat{\varphi}(a_r)$. Conditioned on the result we continue as follows:

   (a) Case $x = a_r = \hat{\varphi}(a_r)$. The state has been projected onto the Hilbert space labelled by one-dimensional irreducible representations $\chi_{m,a_r}$ from box 6.2. Then measure the first register using projectors $P_n := |n\rangle\langle n| \otimes I_A$ with $n \in \mathbb{Z}_p$, getting an outcome $m'$. This finishes the measurement: the final outcome is $m', a_r$ which is the label of a one-dimensional irrep $\chi_{m',a_r}$.
(b) Case \( x = b \neq \hat{\phi}(b) \). In this case we have measured the label of a high-dimensional representation \( \sigma_b \) but the number \( b \) might not be inside our set of representatives \( \hat{R}_b \) (check box 6.2). Therefore, we have to find out the element \( \sigma_{b_r} \) in our set of representatives such that \( \sigma_b \simeq \sigma_{b_r} \). This can be done efficiently computing the representative of the orbit of \( b \in O_{b_r} \), i.e. solving the equation \( b := \hat{\varphi}^{i}(b_r) \). We can do this step efficiently since the group \( \mathbb{Z}_p \ltimes \phi A \) fulfils ARITH.

At the end of the procedure, we have effectively measured the label of an irreducible representation \( \chi_{m,a_r} \) or \( \sigma_{b_r} \).

### 6.3.2 Strong Fourier sampling

Finally, in Strong Fourier sampling one measures first the label of an irrep \( \sigma \) and then one performs an additional measurement over the states corresponding to the matrix coefficients of irrep \( \sigma \). This second measurement is described by projectors \( P_{i,j,\sigma} := |i,j,\sigma\rangle \langle i,j,\sigma| \). It is easy to adapt the scheme we described in the previous section to perform a strong Fourier sampling.

1. Apply the transform \( F \) on \( |(m,a)H\rangle \).

2. Perform a weak Fourier sampling on the state. The output is the label of an irrep \( \chi_{m,a_r} \) or \( \sigma_b \) with \( b = \hat{\phi}^{i}(b_r) \), as we saw in the previous section.

   (a) If \( \sigma = \chi_{m,a_r} \), the irrep is one dimensional and we are done, we have measured the state \( |m,a_r\rangle \).

   (b) If \( \sigma_{b_r} \), then measure the first register using projectors \( P_n := |n\rangle \langle n| \otimes I_A \) with \( n \in \mathbb{Z}_p \), obtaining a number \( m \). The we are done, we have measured the state \( |m,i,b_r\rangle \) which corresponds to a matrix coefficient of \( \sigma_{b_r} \), according to box 6.3.

Regard that in this case we have performed a complete measurement over the whole computational-basis (cf. box 6.3) contrary to weak Fourier sampling.

### 6.4 QFTs for arbitrary choices of basis

In this section, we explain how to use the canonical quantum Fourier transform \( F \) from theorem 6.2.2 to implement other quantum Fourier transforms with different choices of bases for high-dimensional irreducible representations (cf. box 6.2). Essentially, all the ingredients we need are already in box 6.4 where we sketch a short algorithm to change the basis of the irreps in the decomposition of the gate \( Z_R \).

As we now explain, this is enough for our purposes.

Let us consider the non-abelian hidden subgroup problem. Exploiting representation-theoretical symmetries of coset-states [18] the definition of the HSP we saw in chapter 2 can be reformulated as ‘the problem of extracting the generators of the hidden subgroup \( H \) from a quantum state with a density matrix with the following form’.

\[
\rho_H := \bigoplus_{\sigma \in \hat{G}} \frac{1}{|G|} \sum_{h \in H} \sigma(h)^* \otimes I_{ds}
\]  

The quantum-states of the form (6.29) are generated applying a canonical quantum Fourier transform to the original homogeneous distribution of all left-coset states \( \sum |xH\rangle \langle xH| \) and they are used as initial resources to solve the Hidden Subgroup Problem in several quantum algorithms [18]. If we restrict our study to quantum Fourier transforms that help to design quantum measurements for density matrices of the form (6.29) we can exploit that they have the same structure as the unitary matrix \( Z_R \) in box 6.4.

Therefore, the changes of basis for the high dimensional irreps that are relevant to us are exactly those implemented using the algorithm we sketched.

The following theorem described how to implement a new QFT in a different basis using \( F \) and transforming the algorithm explained in box 6.4 into a quantum circuit.
**Theorem 6.4.1.** The unitary gate $F_V := CV \cdot F$ implements a quantum Fourier transforms for the group $\mathbb{Z}_p \rtimes \varphi A$ expressing the high-dimensional irreducible representations of the group $\sigma_{b_r}$ in the basis $V_{\sigma_{b_r}}$, where $F$ is the quantum Fourier transform of theorem 6.2.2 and the gate $CV$ is defined as:

$$CV := \sum_{m,a_r} |m, a_r \rangle \langle m, a_r| + \sum_{m, \hat{\phi}^i (b_r)} V \otimes I_A |n, \hat{\phi}^i (b_r) \rangle \langle n, \hat{\phi}^i (b_r)|$$

(6.30)

The unitary gate $F_V$ can be implemented as the quantum circuit shown in figure 6.2, which is efficient if and only if the circuits of the unitaries $F$ and $V$ are efficient.

![Figure 6.2: QFT in an arbitrary irrep-basis.](image.png)

**Proof:** The correctness of the procedure is explained in box 6.4. Straightforwardly, the circuit is efficient if and only if $F$ is efficient and if we choose a reasonable basis $V$ that can be implemented efficiently as a quantum circuit as well. Notice that, as in theorem 6.2.2, we are imposing that group $\mathbb{Z}_p \rtimes \varphi A$ must fulfill ARITH.

---

### 6.5 Conclusions and relation to previous work

The quantum circuits we have explained in this section are rather flexible allow to implement QFTs for the whole class of groups $\mathbb{Z}_p \rtimes \varphi A$ for arbitrary abelian groups with different choices of basis for the high-dimensional irreducible representations. They can be used to implement efficient non-abelian quantum Fourier transforms for several groups studied in quantum computation, namely the Dihedral Group [13, 14], the wreath products [17], the family $\mathbb{Z}_p \rtimes \mathbb{Z}_{r_p}$ for fixed $r$ which contains the Heisenberg group [43], the family $\mathbb{Z}_2 \rtimes \mathbb{Z}_{n_p}$ studied in [19] and some products of the form $\mathbb{Z}_p \rtimes \mathbb{Z}_{m_p}$ [20]. For all these instances there are efficient quantum algorithms to solve the non-abelian hidden subgroup problem showing exponential quantum speed-ups over the best known classical algorithms. Also, from theorems 6.2.2 and 6.4.1 we can recover some already-known efficient quantum circuits to compute QFTs for the Heisenberg group [36] and the wreath-product [17].

Historically, [36] is one of the first works where efficient quantum circuits to implement non-abelian QFTs were given. There the author found QFTs for some metacyclic groups, including some semi-direct products like the Dihedral Group. In this circuits the idea of using the abelian QFT of a 'big' abelian subgroup $A$ as a subroutine to implement the non-abelian QFT of the whole group was used for the first time in a quantum circuit. This feature is present also in the QFTs given in [37, 17] for some meta-cyclic and some semi-direct groups (the order of the group is restricted to be a power of 2). Our circuits, shown in figures 6.1 and 6.2 also have this structure.

In this work we have not used particular techniques to develop our QFT circuits. Originally, we constructed our quantum circuits generalising a quantum Fourier transform given in [17], finding that the abelian QFT of the abelian direct-product group $\mathbb{Z}_p \times A$ is a particular case of a non-abelian Fourier transform of the semi-direct group $\mathbb{Z}_p \rtimes \varphi A$. However this transform is not ideal for some applications, as it does not block-diagonalise the regular representation of the group in the neat form presented in definition 6.1.1. Instead, it uses a different choice of basis for the irreps for each of the Fourier-transformed regular-representations $\mathbb{Z}_L$ and $\mathbb{Z}_R$. Moreover, in the decomposition $\mathbb{Z}_L$ and $\mathbb{Z}_R$ each copy of a high-dim irrep appears, as well, in a slightly different basis. In the circuits we have presented we have fixed these problems finding proper basis to represent the irreps. Since there are not many available techniques to deal with bases of irreducible representations, the design of the irrep-basis can be a tedious engineering
process, and we do not include this in our discussion. However, our original calculations can be consulted in appendix D.
Chapter 7

Classical simulation of non-abelian quantum Fourier transforms

Summary

In this chapter we investigate the classical simulation of non-abelian quantum Fourier transforms over a class of semi-direct products \( \mathbb{Z}_p \rtimes \phi A \) containing several groups studied in quantum computation \[17, 18, 19, 20\] including the Dihedral group \( D_N \) \[16\]. For these groups we gave efficient quantum circuits to compute their quantum Fourier transforms in the previous chapter (sections \[6.2, 6.4\]). We find that, given any initial coset-state of a group in our class, interesting quantum measurements on the state such as Weak and Strong Fourier sampling can be efficiently simulated in a classical computer.

In the case of Strong Fourier sampling, we also study the effect of the choice of basis of the irreducible representations on the classical simulation. In connection with chapter 6 we find that our classical simulations are always efficient whenever the prime number \( p \) scales as \( O(\text{polylog}|A|) \), independently of the choice of basis. However, in the most general case our techniques per se only allow to simulate choices of basis which are described by sparse matrices. In both regimes we can simulate efficient QFTs for non-abelian groups which are studied in Quantum Computation. \[16, 17, 18, 19, 20\].

The structure of this chapter is the following. In section 7.1 we comment the most relevant features of the groups considered in our classical simulations. In section 7.2 we give a classification of the coset-states of these groups. In section 7.3 we prove some preliminary mathematical results we will use in our simulation scheme. In sections 7.4-7.5 we prove our main classical simulation results. In section 7.6 we comment some particular examples of non-abelian groups for which our simulation results apply.

Conventions

In this chapter all the semi-direct groups \( \mathbb{Z}_p \rtimes \phi A \) fulfil the condition ARITH we saw in the previous section (definition \[6.2.1\]). For these groups the algorithms from chapter 5 and the quantum circuits for the QFTs from chapter 6 are always efficient.

When we consider quantum measurements on \( G \)-states of semidirect product groups \( \mathbb{Z}_p \rtimes \phi A \), the default computational-basis will be \( \{|m,a\} : m \in \mathbb{Z}_p, a \in A \} \). Respectively, for measurements on a \( G \)-state of the abelian group \( A \) (or measurements on the second register of the former basis) the default computational-basis will be \( \{|a\}, a \in A \} \). Apart from these bases, we will be interested in the re-labelled basis of the group \( \mathbb{Z}_p \rtimes \phi A \) summarised in box 6.3. We will only use the latter when we talk about classical simulations of weak and strong Fourier sampling techniques.

The input size of our problems is again \( n := \lceil \log |G| \rceil \) where \( |G| \) is always the order of the group under study. Abelian groups are always presented as cyclic-group decompositions \( A := \mathbb{Z}_{N_1} \times \cdots \times \mathbb{Z}_{N_d} \). By ‘classical simulation’ we always mean ‘efficient classical simulation’.
7.1 Our family of non-abelian groups

In this chapter we only consider semi-direct product groups \( \mathbb{Z}_p \rtimes A \) fulfilling the condition ARITH introduced in the previous chapter, section 6.2.

Our main motivation to study the simulation of non-abelian Fourier transforms over this class of groups is three-fold. The first, they are non-abelian groups and classical simulations of non-abelian QFTs is an interesting problem itself that, to our best knowledge, has not been studied yet. The second, comes from the considerable interest in understanding the complexity of the dihedral hidden subgroup problem due to its applications [14, 16] for quantum computation; our class is big enough to contain this group and several others studied in the field of quantum algorithms [17, 18, 19, 20]. See also the discussion at the end of chapter 2.

The third is practical. In chapter 6 we showed how to implement QFTs for any group of the class ARITH. For the groups inside this class fulfilling ARITH, theorem 6.2.2 states that our circuits for the quantum Fourier transform are efficient and built-up of the following unitary gates:

(a) Abelian quantum Fourier transforms \( F_A \) and their controlled versions \( CF_p \).

(b) Reversible classical gates: controlled-\( X_p \), controlled-\( U_{(-)} \).

(c) Irreducible representation re-labelling operations (cf. box 6.3).

The class ARITH is a simpler case inside our family of semi-direct products since we know that gates of type (c) can be efficiently implemented using reversible classical circuits. This feature and the fact that this family shares some (structural) similarities with abelian groups, gives us a good candidate to investigate the feasibility of simulating non-abelian QFTs with classical computers.

7.2 Quantum coset states

The initial quantum states of the quantum processes we try to simulate classically are arbitrary non-abelian coset-states (cf. equation 2.12) of semi-direct groups of the form \( \mathbb{Z}_p \rtimes A \). From now our semi-direct groups will always fulfil the condition ARITH. We dedicate this section to explain the physical structure and main properties of these coset-states.

7.2.1 The hierarchy of semi-direct coset-states

We define a new family of functions which includes \( \Phi_1 \) (definition 4.4) in order to simplify notation throughout the rest of the work.

**Definition 7.2.1.** For any \( (m,a) \in \mathbb{Z}_p \rtimes A \) define a functions \( f_{ma} \) from \( \mathbb{Z}_p \rtimes A \) to \( A \) as:

\[
f_{ma}^{(n)}(b) := a + \Phi_1^{(n-m)}(\varphi^m(b))
\]

These functions have the same computational cost as \( \Phi_1 \) so they can be efficiently computed classically.

Now, the following lemma gives a complete classification of the coset states of \( \mathbb{Z}_p \rtimes A \). It uses the classification of cosets of \( \mathbb{Z}_p \rtimes A \) we gave in chapter 4 and the functions (7.1) we just defined.

**Lemma 7.2.2.** Any coset-state \( |(m,a)H\rangle \) of \( \mathbb{Z}_p \rtimes A \) can be written either as a uniform disjoint superposition of either 1 or \( p \) coset states of the abelian direc-product group \( \mathbb{Z}_p \times A \). The expressions of these states are:

1. Case \( H = H_A \).

\[
|(m,a)H\rangle = |m\rangle a + \varphi^m(H_A)
\]

2. Case \( H = \langle(1,d),H_A\rangle \)

\[
|(m,a)H\rangle = \frac{1}{\sqrt{p}} \sum_n |n\rangle f_{ma}^{(n)}(d) + H_A
\]
7.2. QUANTUM COSET STATES

Proof: Both equations come straightforwardly using the formulas from box [1.1] \[
\]

As particular cases, any computational-basis state can be written in form 1 in lemma 7.2. For any group of this form the HSP can always be solved efficiently with a quantum algorithm if the hidden subgroup fulfils $H = H_A$, or in other words, if the input of the problem are cosets of type 1. However, despite considerable effort, an efficient algorithm to do the same with the cosets of type 2 has not been discovered, and, if we eventually found it, we could apply it to solve completely the dihedral HSP. A rigorous proof that the HSP over $\mathbb{Z}_p \rtimes_{\varphi} A$ reduces to the case where the subgroup is of type 2 can be found in [33].

7.2.2 Simulating projective measurements on cosets-states

Now we use lemma 7.2.2 to show there exists an ‘adequate’ classical description of any coset-state $(m, a)H$ of the group $\mathbb{Z}_p \rtimes_{\varphi} A$ that allows to simulate quantum projective measurements if the classical description is given to us as an input. This description for a coset state $(m, a)H$ will be an arbitrary poly-size set of generators of $H := \langle h_1, \ldots, h_r \rangle$ and the element $(m, a)$ which identifies the coset. To show this, we prove two lemmas which will be useful throughout the rest of the section. The first of them shows that, for an abelian group $A$ it is possible to simulate any projective measurement over any coset state $|a + H\rangle$ if its ‘adequate’ classical description is given to us.

Lemma 7.2.3. Given any abelian group $A := \mathbb{Z}_{N_1} \times \cdots \times \mathbb{Z}_{N_d}$ and a subgroup $H$ of this group, quantum projective measurement on the computational-basis acting on an coset state $|a + H\rangle$ can be efficiently simulated classically if a poly-size generating-set $S := \{h_1, \ldots, h_r\}$ of the subgroup $H$ and the representative element $a$ of the coset are given as an input.

Proof: Any coset state $|a + H\rangle$ is a uniform superposition over the states belonging to the coset $a + H$. Hence, simulating a projective measurement on the basis $|n, b\rangle$ is equivalent to sampling a random element from the support of the quantum state, which can be done sampling elements $h$ of the group with uniform probability and then adding the shift $a + h$. To sample uniformly from the group we use algorithm (a) from theorem 5.2.2.

Now, we prove a small lemma which shows that if it is possible to simulate projective measurements on over a family of $\{|\psi_n\rangle\}$ of $n$-qubit quantum states, then it is also possible to simulate projective measurements over uniform disjoint superpositions of them.

Lemma 7.2.4. Given a family $n$-qubit quantum state of $\{|\psi_m\rangle\}_{m=0}^{N-1}$ living on the Hilbert space $\mathcal{H}$, where the number of elements in the family is $N = O(2^n)$ and each state $|\psi_m\rangle$ has a classical description that can be used by a classical computer in order to efficiently simulate projective-measurements on the state in a fixed computational-basis $\mathcal{B}_{\mathcal{H}}$. Then, given an efficient classical subroutine to compute the classical description of each state $|\psi_m\rangle$ and a uniform quantum superposition of the form

$$|\alpha\rangle := \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} \alpha_m |m\rangle |\psi_m\rangle$$ (7.4)

where all the coefficients have equal modulus $\|\alpha_m\|^2 = 1$, then quantum projective measurements on the basis $\{|m\rangle\}_{m=0}^{N-1} \otimes \mathcal{B}_{\mathcal{H}}$ can be efficiently simulated classically in time $O(\text{poly}(n))$.

Proof: Because we can always measure the first register first without loss of generality, the classical simulation can be done simulating first this measurement and then simulating a measurement on a state of the form $|\psi_m\rangle$ conditioned on the outcome of the first measurement. Regard that the coefficients $\alpha_m$ do not play a role in this scheme since $\|\alpha_m\|^2 = 1$ implies that they do not contribute to the probability of the outcomes if we measure the first register on the basis $|m\rangle$.

To simulate the first measurement we sample a random number from 0 to $N-1$ getting an outcome $m$ in time $O(\text{poly}(n))$. Then we ask the given subroutine for the ‘adequate’ classical description of the state $|\psi_m\rangle$. Using the description we can simulate a projective measurement over such state in the basis $\mathcal{B}_{\mathcal{H}}$. The probability distribution of the outcomes it is the same as if we had done the quantum measurement on the basis $\{|m\rangle\}_{m=0}^{N-1} \otimes \mathcal{B}_{\mathcal{H}}$ directly. \]
Finally, we can combine the two previous lemmas to show how to simulate a measurement on the basis \(|m,a⟩\) for any coset-state of a semi-direct group \(\mathbb{Z}_p \rtimes ϕ A\).

**Theorem 7.2.5.** Given a coset state \(|(m,a)H⟩\) of a semi-direct product \(\mathbb{Z}_p \rtimes ϕ A\), quantum projective measurements on the computational-basis acting on this state can be efficiently simulated classically if a poly-size generating-set \(S_H = \{h_1, \ldots, h_r\}\) of the subgroup \(H\) and the representative element of the coset \((m,a)\) are given as an input.

**Proof:** From the classification lemma 7.2.2 it follows that semi-direct coset states are uniform superpositions \(\sum |n⟩|ψ_n⟩\) of abelian coset states \(|ψ_n⟩= |a_n + H_n⟩\). For an abelian coset of the form \(|a_n + H_n⟩\), given a classical description \(\{S_n, a_n\}\) where \(S_n\) is a generating-set of \(H_n\), lemma 7.2.3 shows how to simulate quantum measurements on the computational-basis acting on these states. Hence, it suffices to show that for any semi-direct coset state we can compute the classical descriptions \(\{S_n, a_n\}\) given \((m,a)\) and a generating set \(S_H\) of \(H\). Then, we can use the scheme given in lemma 7.2.4 to simulate the measurements.

In the case \(H = H_A\) the state is just \(|(m,a)H_A⟩ = |m⟩|a + ϕ^m(H_A)⟩\). The classical description \(\{S_m, a_m\}\) is \(\{S_m, a_m\} = \{ϕ^m(S_H), a\}\) which can be efficiently computed since automorphisms can be efficiently computed. In the case \(H \neq H_A\), using the algorithm form of lemma 4.3.2 we compute from \(S_H\) a new set of generators of the group \(H\) in canonical form \(S'_H = \{(1,d), S_A\}\), where \(S_A\) generates \(H_A\). From equation 7.3 it follows that the classical descriptions we need to compute are of the form \(\{S_n, a_n\} = \{S_A, f_{n,a}(d)\}\). Because the functions \(f_{n,a}\) are efficiently computable, the description is efficiently computable.

### 7.3 Preliminaries

In this section we prove some preliminary mathematical results that will be used in section 7.4 to prove our main theorems.

#### 7.3.1 Decomposition of the circuit \(F\)

Our first step will be to decompose the circuit \(F\) into two smaller quantum circuits which perform operations with distinct representation-theoretical interpretation (figure 7.1). A consequence of theorem 6.4.1 (cf. figure 6.2) is that the gate \(CU^\hat{R}\) in the quantum circuit of \(F\) is performing a change of basis of the high-dimensional irreducible representations. As a result, this gate is irrelevant in weak Fourier sampling since a property of this measurement scheme is that it is always independent of the choice of basis of the irreducible representations [25]. For our groups, this can be seen simply from the scheme shown in section 6.3.1 since the part of the Hilbert space being measured is not affected by the action of the gate \(CU^\hat{R}\).

![Figure 7.1: Decomposition of the QFT circuit](image)

As a result, it makes sense to simulate the action of the two first gates of the circuit \(F\) independently, defining a new gate

\[
F_W := CF_p \cdot (I_p \otimes F_A)
\]  

(7.5)

Where the name \(F_W\) is chosen to remind that any QFT in any arbitrary basis used to implement weak Fourier sampling over coset-sates of the group \(\mathbb{Z}_p \rtimes ϕ A\) generates the same probability distribution of
outcomes as $F_W$. In terms of $F_W$ the circuit $F$ can be re-written as

$$F := CU_{R^*}F_W$$ (7.6)

### 7.3.2 Simulation of abelian cosets

Before showing how to simulate the action of a non-abelian quantum Fourier transform on a non-abelian coset it is important to understand how can the same task be accomplished in the abelian case, which is the content of the next lemma.

**Lemma 7.3.1.** Given an abelian group $A := \mathbb{Z}_{N_1} \times \ldots \times \mathbb{Z}_{N_d}$ and a quantum state $|\psi\rangle := F_A|a + H\rangle$ obtained after applying the abelian quantum Fourier transform $F_A$ on a coset state of the group $|a + H\rangle$, then quantum projective measurements in the computational-basis acting on the state $|\psi\rangle$ can be efficiently simulated classically if a poly-size generating-set $S_H := \{h_1, \ldots, h_r\}$ of the subgroup $H$ and the representative element $a$ of the coset state are given as an input.

**Proof:** Recalling from the algorithm to solve the abelian-HSP (section 2.2) that the output state after applying the quantum Fourier transform $F_A$ has the following expression

$$|\psi\rangle = \frac{1}{\sqrt{|H|}} \sum_{g \in H^*} \chi_g(a_1) |g\rangle$$ (7.7)

If we measure this state we obtain an element $g$ from the orthogonal subgroup $H^*$ with equal probability and the problem reduces to sampling from the orthogonal subgroup with uniform probability given a system of generators $S_H$ of the group $H$. But this problem can be solved efficiently using an algorithm we gave in theorem 5.2.2, so we are done. □

### 7.3.3 An important lemma

Finally, we prove a lemma that shows that some characters functions of $A$ that will appear later in the text can be replaced with $p$-th roots of the unity in certain situations. Here we use some of the definitions we introduced in chapter 4.

**Lemma 7.3.2.** Given a subgroup $H = \langle (1,d), H_A \rangle$ of the semi-direct group $\mathbb{Z}_p \rtimes \varphi A$ and an element $g \in H_A^*$ such that $g = \hat{\varphi}(g)$. Then $\chi_g(d)$ is a $p$-th root of the unity and we define a number $m^*_g$ such that

$$\chi_g(d) = \omega^{m^*_g}$$ (7.8)

Where we omit the dependence of $m^*_g$ on $d$ since this element will always be fixed.

**Proof:** $(1,d)^p = (0, \Phi_A^p(d))$ belongs to $H_A$, so it must hold

$$\chi_g(\Phi_A^p(d)) = 1$$ (7.9)

Now expand $\Phi_A^p = d + \varphi(d) + \ldots + \varphi^{p-1}(d)$ and use the linearity of the character functions to get:

$$1 = \chi_g(\varphi(d)) \ldots \chi_g(\varphi^{p-1}(d)) = \chi_g(d)^p$$ (7.10)

The second equality holds since $\chi_g(\varphi(d)) = \chi_{\hat{\varphi}^{-1}(g)}(d)$ and $g = \hat{\varphi}(g)$, implying that all the characters in the above product are equal to $\chi_g(d)$. □
7.4 Non-abelian Fourier transforms can be classically simulated

In this section we give our first main result, which shows that if the quantum circuit to compute the canonical quantum Fourier transform $F$ given in theorem 6.2.2 is applied to arbitrary coset-states $|(m, a)H\rangle$ of $\mathbb{Z}_p \rtimes \varphi A$ then we can simulate the action of the circuit on the state step-by-step, meaning that a measurement on the computational basis performed immediately after the action of any of the quantum gates involved in this circuit $(F_A, CF_p$ and $CU_R)$ can be simulated efficiently in a classical computer.

In our result there are no restrictions on the subgroup $H$ that defines the coset-states, it could be trivial, abelian, normal or even a subgroup of the form $Fg$ gates involved in this circuit ($F_A$, $CF_p$ and $CU_R$) can be simulated efficiently in a classical computer.

As a consequence, the quantum circuit shown in theorem 6.2.2 can be efficiently simulated classically step-by-step for the given inputs.

As a consequence, the quantum circuit shown in theorem 6.2.2 can be efficiently simulated classically step-by-step for the given inputs.

Case $H = H_A$, defining $H' := \varphi^m(H_A)$ and compute (efficiently) a new set of generators $S'_H = \varphi^m(S_H)$. The final state is

$$|\psi_A\rangle = |m\rangle F_A|a + H'\rangle = |m\rangle \left( \sum_{g \in H'} \frac{1}{\sqrt{|H'|}} \chi_g(a)(g) \right)$$

The classical simulation of the measurement on the final state can be done efficiently because of lemma 7.3.1.

Case $H = \langle (1, d), H_A \rangle$. Using that the second register always contains an abelian coset state and equation 7.7 we obtain the final state.

$$|\psi_A\rangle = \frac{1}{\sqrt{|p| H_A^\perp}} \sum_n |n\rangle \otimes \sum_{g \in H_A^\perp} \chi_g(f_{na}^m(d)) |g\rangle$$

$$= \frac{1}{\sqrt{|p| H_A^\perp}} \sum_{g \in H_A^\perp} \left( \sum_n \chi_g(f_{na}^m(d)) |n\rangle \right) \otimes |g\rangle$$

Proof: We prove (a),(b) and (c) in order. In each case, one must prove that the statement holds for both types of coset-states given in lemma 7.3.

Proof of (a) We want to show that measurements on $|\psi_A\rangle$ can be simulated.

- Case $H = H_A$, defining $H' := \varphi^m(H_A)$ and compute (efficiently) a new set of generators $S'_H = \varphi^m(S_H)$. The final state is

$$|\psi_A\rangle = |m\rangle F_A|a + H'\rangle = |m\rangle \left( \sum_{g \in H'} \frac{1}{\sqrt{|H'|}} \chi_g(a)(g) \right)$$

- Case $H = \langle (1, d), H_A \rangle$. Using that the second register always contains an abelian coset state and equation 7.7 we obtain the final state.

$$|\psi_A\rangle = \frac{1}{\sqrt{|p| H_A^\perp}} \sum_n |n\rangle \otimes \sum_{g \in H_A^\perp} \chi_g(f_{na}^m(d)) |g\rangle$$

$$= \frac{1}{\sqrt{|p| H_A^\perp}} \sum_{g \in H_A^\perp} \left( \sum_n \chi_g(f_{na}^m(d)) |n\rangle \right) \otimes |g\rangle$$
The first equation shows that the outcomes of the projective measurement are elements \((n, g)\) where \(n\) is a random number from 0 to \(p - 1\) and \(g\) is an element of the orthogonal subgroup \(H_{A}^{\perp}\) taken with uniform probability. Hence, the simulation reduces to sampling from the orthogonal subgroup using the algorithm from theorem 5.2.2.

\[\text{Proof of (b)}\]

Now we show how simulate measurements on \(|\psi_{W}\rangle := CF_{p} |\psi_{A}\rangle\). From the definition of \(CF_{p}\) it makes sense to separate in two parts: one supported on those states \(|n, g\rangle\) with \(g \neq \hat{\varphi}(g)\) and the second supported on the complementary.

- **Case** \(H = H_{A}\) and \(g \in H_{A}^{\perp}\)

\[
|\psi_{A}\rangle = \frac{1}{\sqrt{|H_{A}^{\perp}|}} \left( |m\rangle \otimes \sum_{g = \varphi(g)} \chi_{g}(a)|g\rangle + |m\rangle \otimes \sum_{g' \neq \varphi(g')} \chi_{g'}(a)|g'\rangle \right) \quad (7.17)
\]

\[
\Rightarrow |\psi_{WF}\rangle = \frac{1}{\sqrt{|H_{A}^{\perp}|}} \left( F_{p}|m\rangle \otimes \sum_{g = \varphi(g)} \chi_{g}(a)|g\rangle + |m\rangle \otimes \sum_{g' \neq \varphi(g')} \chi_{g'}(a)|g'\rangle \right) \quad (7.18)
\]

(7.19)

To simulate a measurement, we can w.l.o.g assume that we measure first the second register. Since \(CF_{p}\) leaves this part of the state untouched, the outcome of this measurement is again an element of the orthogonal \(H_{A}^{\perp}\) taken with uniform probability, and we simulate this, again, using the algorithms from theorem 5.2.2 to compute a random element from the orthogonal \(g\). In a real measurement, if the outcome \(g\) of this process fulfills \(g \neq \hat{\varphi}(g)\) the first register of the projected state contains a fix value \(m\). We simulate this returning this value. If \(g \neq \hat{\varphi}(g)\) the first register of the projected state would be \(|\psi_{p}\rangle := F_{p}|m\rangle\) which is a uniform superposition over the states \(|n\rangle\) with \(n\) from 0 to \(p - 1\). We simulate the projective measurement by computing a random number in \(\mathbb{Z}_{p}\) and returning it.

- **Case** \(H = \langle (1, d), H_{A}\rangle\) and \(g \in H_{A}^{\perp}\).

From now we work with un-normalised states and remove the factors \(1/\sqrt{|H_{A}^{\perp}|}\), since it is obvious from the previous examples that we do not need to compute them to simulate quantum measurements. We split the state \(|\psi_{A}\rangle\) as in the previous case.

\[
|\psi_{A}\rangle \propto \sum_{g = \varphi(g)} \sum_{n} \left( \chi_{g}(f_{ma}^{(n)}(d))|n\rangle \otimes |g\rangle + \sum_{g' \neq \varphi(g')} \sum_{n} \left( \chi_{g'}(f_{ma}^{(n)}(d))|n\rangle \otimes |g'\rangle \right) \quad (7.20)
\]

In the case \(g = \hat{\varphi}(g)\) we can use the following facts to simplify the term \(\chi_{g}(f_{ma}^{(n)}(d))\)

1. \(\chi_{g}(\Phi_{1}^{m}(d)) = \varphi_{m}^{n*}\), as a consequence of lemma 7.3.2
2. \(\Phi_{1}^{n-m}(\varphi_{m}(d)) = \Phi_{1}^{m}(d) - \Phi_{1}^{m}(d)\), by definition.

Resulting on

\[
\chi_{g}(f_{ma}^{(n)}(d)) = \chi_{g}(a') \varphi_{m}^{n*}\quad (7.21)
\]

Where \(a' = a - md\). From this equation one obtains the equality

\[
\sum_{n} \chi_{g}(f_{ma}^{(n)}(d))|n\rangle = \sqrt{p} \chi_{g}(a')F_{p}^{|m_{g}^{*}\rangle}|g\rangle \quad (7.22)
\]

And applying \(CF_{p}\) the inverted Fourier transform \(F_{p}^{|m_{g}^{*}\rangle}\) cancels:

\[
CF_{p} \sum_{n} \chi_{g}(f_{ma}^{(n)}(d))|n\rangle = \sqrt{p} \chi_{g}(a')|m_{g}^{*}\rangle|g\rangle \quad (7.23)
\]
This equality can be used to obtain the following expression for the state $|\psi_{\mathcal{F}_W}\rangle$.

$$|\psi_{\mathcal{F}_W}\rangle = CF_p|\psi_A\rangle$$

$$\propto \sum_{g=\varphi(g)} \sqrt{p}\chi_g(a')|m^*_g,g\rangle + \sum_{g' \neq \varphi(g')} \sum_n \left(\chi_{g'}(f'_{m,a}^n(d))|n\rangle\right) \otimes |g'\rangle$$

This transformation has a group-theoretical interpretation: the gate $CF_p$ acts on the states $g = \varphi(g)$ and applies the right interference to obtain the computational-basis-states $|m^*_g,g\rangle$ that labels the irreducible-representations $\chi_{m^*_g,g}$ which are exactly the ones equivalent to the identity when they are restricted to the subgroup $H_A$.

As in the previous case, $CF_p$ does not affect the second register, so we can simulate a measurement of this part of the state computing a random element $g$ from an orthogonal subgroup using the algorithms from theorem 5.2.2. In a real measurement, if $g \neq \varphi(g)$ equation (7.25) tells us that the state would be projected to a uniform superposition over all states $|n\rangle$ and to simulate a measurement of this register we compute a number from 0 to $p-1$. On the other hand, if $g = \varphi(g)$ the state would be projected to a product state $|m^*_g,g\rangle$ and we need to return the number $m^*_g$ which would be the outcome of the quantum measurement. The explicit formula given in lemma 7.3.2 allows us to compute this number using modular arithmetic, since we know $d, g$ and $p$. Therefore, we are done with part (b).

Proof of (c)

Finally, it is straightforward how to simulate $\mathcal{F}$ from the previous case, using the formula $\mathcal{F} = CU_p \mathcal{F}_W$. Theorem 6.2.2 tells us that $CU_p$ is just a efficiently computable classical circuit due to the third condition of $ARITH$, or in other words, just an efficiently computable permutation of the original computational-basis. Then, to simulate a quantum measurement on the computational basis of $\mathcal{F}[(m, a)H] = CU_p \mathcal{F}_W[(m, a)H]$ it is enough to simulate a quantum measurement on $\mathcal{F}_W[(m, a)H]$, giving outcomes $(n, g)$ and then apply the permutation $CU_p$ to the outcomes.

7.5 Simulating weak and strong Fourier sampling

In this section we prove our second main result, showing that for the non-abelian groups $\mathbb{Z}_p \ltimes \varphi A$ the quantum measurements known a weak and strong Fourier sampling performed over arbitrary coset states $|(m, a)H\rangle$ of the group can be efficiently simulated on a classical computer. In the case of strong Fourier sampling we will take into account the change of basis of the irreducible representations. In this section we use the more general quantum Fourier transforms $\mathcal{F}_V$ from theorem 6.4.1 that allow to represent the high-dimensional irreducible representations of the group $\sigma_b$, in alternative basis $V\sigma_b, V^\dagger$.

7.5.1 General case

First we considered the general case $\mathbb{Z}_p \ltimes \varphi A$. The following result shows that, for any group of our family weak Fourier sampling can always be simulated, and that strong Fourier sampling can be simulated for a class of non-abelian quantum Fourier transforms $\{\mathcal{F}_V\}$ which corresponds to the transforms of theorem 6.4.1 which implement changes of basis of the irreducibles that are $p \times p$ monomial matrices.

Theorem 7.5.1. Given an arbitrary coset state $|(m, a)H\rangle$ of a semi-direct product $\mathbb{Z}_p \ltimes \varphi A$, if a poly-size set of generators $S_H$ of the subgroup $H$ and the representative element of the coset $(m, a)$ are given as an input, then:

(a) Weak Fourier sampling the state $|(m, a)H\rangle$ using any quantum Fourier transform of the group can be efficiently simulated classically.
7.5. SIMULATING WEAK AND STRONG FOURIER SAMPLING

(b) **Strong Fourier sampling** the state $|\langle m, a \rangle_H\rangle$ using the canonical quantum Fourier transform $F$ from theorem 6.2.2 can be efficiently simulated classically.

(c) **Strong Fourier sampling** the state $|\langle m, a \rangle_H\rangle$ using any quantum Fourier transform $F_V$ from theorem 6.4.1 can be efficiently simulated classically if the choice of basis of the irreducible representations $V$ is an efficiently computable monomial matrix.

**Proof:** In fact, it is not hard to see that (c) ⇒ (b) ⇒ (a) so we could prove just the third one. However, we find more illustrative to prove the three propositions in order. Also, remind that a unitary matrix is monomial if it has one entry per row and column

**Proof of (a)** The first part of the theorem comes from the fact that weak Fourier sampling is independent of the choice of basis of the irreps. In other words, if we apply any Fourier transform $F_V$ on the coset state $|\langle m, a \rangle_H\rangle$ and we apply the measurement scheme we gave in section 6.3.1 to perform weak Fourier sampling for the groups $\mathbb{Z}_p \ltimes \varphi A$, it follows that the gates $CU_P$ and $CV$ from theorems 6.2.2 and 6.4.1 act on a corner of the Hilbert space which is ignored by the measurement scheme. Hence, the statistics are the same as if we had applied the Fourier transform $F_V$.

Now we use theorem 7.4.1 to simulate a computational-basis measurement on the state $F_W|\langle m, a \rangle_H\rangle$. According to the procedure given in section 6.3.1 if we group the outcomes into boxes using the following rules we simulate weak Fourier sampling statistics:

- Whenever we measure a basis state $|n, a_r\rangle$ with $a_r = \hat{\varphi}(a_r)$ this corresponds to the label of a one-irreducible representation $\chi_{n, a_r}$ which we count a valid outcome of weak Fourier sampling.

- On the other hand, if we measured a state $|n, b\rangle$ with $b \neq \hat{\varphi}(b)$ we know we have measured a re-labelled computational state of the form $|n, i, b_r\rangle$ where $b_r$ is some choice of representative of the orbit $O_{b_r}$ of $b$ (a convenient choice in this algorithm would be to take the first element sampled from a particular orbit). Since the group fulfills ARITH we can compute the representative $b_r$ and we know that we have measured the label of an irrep $\sigma_{b_r}$ which we count as a valid outcome of weak Fourier sampling. □

**Proof of (b-c)** Now we will show how to simulate Strong Fourier sampling for groups $\mathbb{Z}_p \ltimes \varphi A$ following the measurement scheme we gave in section 6.3.2. In this case we use theorem 7.4.1 to simulate a measurement on the computational basis over the state $F|\langle m, a \rangle_H\rangle$. We get outcomes which we re-label properly using the rules of box 6.3 which is enough to simulate strong Fourier sampling for the transform $F$ and prove (b). Now remember the circuit of $F_V$ from theorem 6.4.1 and the expression of $CV$.

$$CV := \sum_{m, a_r} |m, a_r\rangle\langle m, a_r| + \sum_{m, \hat{\varphi}^r(b_r)} V \otimes I_A |n, \hat{\varphi}^r(b_r)\rangle\langle n, \hat{\varphi}^r(b_r)|$$

If the matrices $V$ are classically efficiently computable monomial unitaries then $CV$ is a classical efficiently computable monomial unitary and can be written like the product of a diagonal gate and a permutation $CV := DP$. A measurement on the re-labelled computational-basis from 6.3 cancels the relative phases added by the diagonal gate $D$ and only sees the effect of the permutation $P$. Then the simulation of strong Fourier sampling with $F_V$ can be done as follows: simulate strong Fourier sampling with $F$, get an outcome $|i, j, \sigma\rangle$ and transform the labels of this outcomes according to the permutation $P$ defined by $CV$.

### 7.5.2 Groups with small $p$

We finish this chapter showing that it is possible to get even stronger results than theorem 7.5.1 for groups $\mathbb{Z}_p \ltimes \varphi A$ where the prime number $p$ is ‘small’ compared to the order of the abelian group $|A|$. More rigorously if the asymptotic growth of $p$ is at most poly-logarithmic $p = O(\log|A|)$ then we can show that Strong Fourier sampling can be simulated for any coset-state $|\langle m, a \rangle_H\rangle$ and any quantum Fourier transform $F_V$ of the group independently of the choice of basis of the irreps.
Corollary 7.5.2. For any semi-direct product \( \mathbb{Z}_p \ltimes \varphi A \) with \( p = O(\text{polylog}|A|) \) and any quantum Fourier transform \( F_V \) from theorem 6.4.1, Strong Fourier sampling the state \( |(m, a)H\rangle \) can be efficiently simulated classically if a poly-size set of generators \( S_H \) of the subgroup \( H \), the representative element of the coset \((m, a)\) are given as an input, and if the choice of basis of the irreps \( V \) is an efficient quantum circuit.

Proof: Briefly, if \( p \) is small we can apply brute force to simulate the measurement: we show this in detail. If we go back to last expressions of the proof of theorem 7.3.1 we see that it is enough to show that a quantum projective measurement over the state \( CVF|{(m, a)H}\rangle \) can be simulated for any coset, where the \( CV \) gate is.

\[
CV := \sum_{m,a} |m, a\rangle \langle m, a| + \sum_{m} V \otimes I_A |\hat{\psi}^i(b_m)\rangle \langle n, \hat{\psi}^i(b_m)|
\]  

As the gates \( CF_p \) and \( CU_R \) the gate \( CV \) is a controlled gate on the abelian register of the quantum states. Hence, a measurement of \( CV \cdot F|{(m, a)H}\rangle \) on the computational basis is equivalent to measuring the abelian register, performing a controlled operation on the first register conditioned on the outcome and then measuring it. As in the proof of theorem 7.3.1 we simulate the measurement on the second register computing, again, a random \( g \) element from the orthogonal of a subgroup whose elements we know. In the real measurement we would have projected the state into \( |\psi_g\rangle \) where the quantum state \( |\psi_g\rangle \) is a superposition over the states \( |n\rangle \) with \( n \) from 0 to \( p - 1 \) which can be explicitly computed from the formulas of the proof of theorem 7.3.1 using \( O(\text{poly}(p)\text{polylog}|A|) \) time and memory resources. Now, the circuit we have to apply conditioned on the obtained outcome \( g \) is: the identity if \( g = \hat{\psi}(g); V \otimes I_A \) if \( g \neq \hat{\psi}(g) \). In any case we apply this circuit on the projected state and compute explicitly the final state of the computation \( |\psi_g\rangle'\langle g| \). The computational cost, dominated by the \( p \times p \) matrix multiply is \( O(\text{poly}(p)\text{polylog}|A|) \). Now, to simulate a measurement on the first register we can compute from \( |\psi_g\rangle'\langle g| \) all the final probabilities of the different outcomes, store them and use this information to simulate the measurement. The computational cost is again \( O(\text{poly}(p)\text{polylog}|A|) \). Since now that \( p \) is exponentially small compared to \( |A| \), at the end we use \( O(\text{polylog}|A|) \) time and memory resources, so the approach is efficient.

7.6 Examples

We end this chapter showing that our simulation results can be applied to several interesting examples of groups \( \mathbb{Z}_p \ltimes \varphi A \) taken from the non-abelian hidden subgroup problem literature and related to exponential quantum speed-ups.

- The following groups have ‘small’ \( p \) and, for all of them, the classical simulation of Strong Fourier sampling is possible for the big class of quantum Fourier transforms given in corollary 7.5.2: the dihedral group \( \mathbb{Z}_2 \ltimes \mathbb{Z}_N \) [13] [16] [15], the wreath products \( \mathbb{Z}_2 \ltimes \mathbb{Z}_p^{2n} \) from [17], the class \( \mathbb{Z}_2 \ltimes \mathbb{Z}_p^n \) from [19] and the family \( \mathbb{Z}_p \ltimes \mathbb{Z}_p^n \) from [20]. All these groups fulfil the condition ARITH (cf. appendix 9). Except for the dihedral group, the HSP can be solve efficiently in quantum polynomial time for these groups.

- In the general case, theorem 7.5.1 shows that both weak Fourier sampling and strong Fourier sampling for a class of quantum Fourier transforms (smaller than in the previous case) can be efficiently simulated classically. Obviously, all groups of the previous case also fall in this category. A family of groups \( \mathbb{Z}_p \ltimes \varphi A \) which does not have ‘small’ \( p \) but fulfils ARITH are the groups \( \mathbb{Z}_p \ltimes \mathbb{Z}_p^r \) for fixed \( r \) for which an efficient quantum algorithm for the hidden subgroup problem is given in [13] (cf. the original reference for more details). For \( r = 2 \), this group is isomorphic to a group of Pauli matrices, as we mentioned in chapter 2.
Chapter 8
Conclusions and open questions

Overview

In this thesis we have showed that the classical simulation of quantum Fourier transforms for certain instances of non-commutative groups can be achieved, proving that, for a family of non-abelian groups $\mathbb{Z}_p \rtimes \phi A$, it is possible to efficiently simulate in a classical computer the quantum measurements based on non-abelian QFTs known as Weak and Strong Fourier sampling when they are performed on particular classes of initial quantum states. The initial quantum states we consider are not only computational-basis states (product states) but arbitrary coset state $| (m, a) H \rangle$ of the group $\mathbb{Z}_p \rtimes \phi A$. These states are interesting since they can be arbitrarily large quantum superpositions and because they are used in quantum algorithms to solve the hidden subgroup problem. In particular, our classical simulation results apply to initial computational-basis states, abelian-subgroup cosets, normal-subgroup cosets and the coset-states for which the hidden subgroup problem over $\mathbb{Z}_p \rtimes \phi A$ is more difficult to solve [43]. In our simulation we use an adequate classical description of these states in terms of generating-sets of subgroups. Using these descriptions, the simulation can be carried out using classical algorithms to solve computational problems over finite groups.

To prove our main results we have also constructed new quantum circuits to implement quantum Fourier transforms for semi-direct groups of the form $\mathbb{Z}_p \rtimes \phi A$. Our circuits are quite flexible and allow to use different choices of basis for the irreducible representations.

Our work is related to preceding results concerning classical simulation of abelian quantum Fourier transforms, such as the Gottesman-Knill theorem [4] and [11, 12, 13]. To our best-knowledge, this is the first study concerning classical simulations of non-abelian QFTs at present date.

Open questions

From our observations some questions which remain open can be posed. In first place, in our work we have restricted to a class of non-abelian groups studied in the hidden subgroup literature whose quantum Fourier transforms are known to have applications in Quantum Computation and that we consider of interest since it contains hard-instances of the HSP. In particular, the Dihedral Group belongs to this family, whose quantum computational complexity is still unknown.

However, there are other interesting non-abelian groups we have not considered and for which the hidden subgroup problem has been investigated; it could be interesting to study classical simulation of QFTs for some of them. Still, as we have commented in chapter [2] one has to make careful choices since the role of non-abelian quantum Fourier transforms in quantum algorithms is worse understood than in the abelian case and still topic of research. A reasonable next-step could be considering classical simulations of quantum Fourier transforms for groups of the form $H \rtimes A$ where both factors are abelian, like the ones studied in [17].

Also, inside the hidden subgroup problem literature new sampling techniques beyond Fourier sampling have been recently investigated. In particular, a series of relatively recent works [31, 33, 35, 62, 47] suggest
that Clebsch-Gordan transforms can be combined with Fourier transforms in order to implement highly-entangled measurements over multiple registers that help to solve some instances of the hidden subgroup problem. It would be interesting to study the classical simulation of these transforms for the class of semi-direct groups we have introduced $\mathbb{Z}_p \rtimes \phi A$.

As in the abelian case \cite{11, 12, 13} our work does not imply that algorithms showing exponential quantum speed-ups can be efficiently simulated classically. As it is discussed in \cite{11}, these results could perhaps be applied to study other quantum circuits that use quantum Fourier transforms as a subroutine. Although it remains for future works to explore this possibility, we can conclude from these investigations that there is no reason to believe that the appearance of quantum Fourier transforms in quantum circuits is a fundamental obstacle towards their efficient classical simulation.
Part IV

Appendixes
Appendix A

Representation Theory

In this appendix we show how to obtain a complete set of irreducible representations of the semi-direct group $\mathbb{Z}_p \rtimes \phi A$ where $p$ is a prime number and $A$ is an abelian group. The main tool of this chapter is the little group method of Wigner and Mackey, a group-theoretical tool to classify the irreducible representations of semi-direct products of the form $H \rtimes A$ when $A$ abelian. Our presentation is oriented to quantum physics notation and follows the exposition from Serre’s book [60].

We use the notation and definitions of chapter [4]. Our semi-direct products are defined with the ‘white’ multiplication rule (4.1.1).

A.1 Little group method

We will first expose the general method and then apply it to the case where $H$ is cyclic of prime-order $p$.

A.1.1 Induced representations

We first remember the definition of induced representations.

**Definition A.1.1 (Induced Representation).** Let there be a representation $\rho$ of dimension $d$ of a subgroup $H$ of $G$, $H$ with index $n = |G/H|$ and $R$ a list of left-representatives of the quotient set $G/H = \cup_{r \in R} r \cdot H$. The induced representation $\rho^G_H$ is a new representation of $G$ defined as follows

$$\rho^G_H (g) = \sum_{r,s \in R} \hat{\rho}(r^{-1} \cdot g \cdot s) |r \rangle \langle s|$$

(A.1)

Where $\hat{\rho}(x) = \rho(x) \cdot \delta(x \in H)$ is a $d \times d$ matrix $\rho(x)$ when $x := r^{-1} \cdot g \cdot s$ belongs to $H$ and the zero matrix $0_d$ of dimension $d$ if not. Thus, the induced representation $\rho^G_H$ has dimension $nd = d \times |G/H|$.

A.1.2 The little group action

Now consider the group $G := H \rtimes A$, where $A$ is abelian and $H$ is a group of automorphisms of $A$ with generators $S_H := \{ \varphi_1, \ldots, \varphi_n \}$, i.e., the elements of $H$ are automorphisms $\phi$ that obtained composing the generators $\varphi_i$. The main tool used by little group method is a group action from $H$ into the character group $\hat{A}$ of the abelian group $A$.

**Proposition A.1.2.** For any automorphism $\phi$ from the group $H$ the function $\hat{\phi} : \hat{A} \rightarrow \hat{A}$

$$\chi_{\hat{\phi}(\alpha)} := \chi_\alpha \circ \phi^{-1}$$

(A.2)

is also a group automorphism of the dual group $\hat{A}$ which defines a group action from $H$ into $\hat{A}$ via

$$\phi \cdot (\chi_\alpha) := \chi_{\hat{\phi}(\alpha)}$$

(A.3)
**Proof:** Let’s prove first that the action is indeed an action. The application \( \chi_a \circ \phi^{-1} \) is well-defined since composition of two homomorphisms is a homomorphism and zero-dimensional representations are always irreducible. Almost immediate are: \( \chi_{\phi \circ \phi(a)} = \chi_{\phi(\phi(a))} \) and \( \chi_{\phi a} = \chi_a \), hence \( \phi \) defines an action. Any action defines a bijective map on \( \hat{A} \) and in this case, also a homomorphism since for \( \chi_a \cdot \chi_a' = \chi_{a + a'} \) we have \( \chi_{\phi(a+a')} = (\chi_a \cdot \chi_{a'}) \circ \phi^{-1} = \chi_{\phi(a)} \cdot \chi_{\phi(a')} \) for any \( g \) from \( A \).

The little group method makes crucial use of this action to define a partition of \( \hat{A} \) into orbits of \( \hat{\phi} \) or (disjoint invariant spaces under \( \hat{\phi} \)). Since the character group \( \hat{A} \) is isomorphic to \( A \) via \( a \leftrightarrow \chi_a \), then \( \hat{\phi} \) is also a group automorphism of \( A \) and the partition of the character group induces a partition of the group \( A \). Since it is a common habit to index character functions \( \chi_a \) using elements of the original abelian group \( A \) we will work with the partition of \( A \) for clarity.

### A.1.3 Method

Using the \( \hat{\phi} \) partition of the group \( A \), the little group method ‘assigns’ an inequivalent irreducible \( \sigma_a \) of \( G \) to each orbit \( O_a \) of the partition in a one-to-one way. Therefore, classifying these orbits characterises all possible irreducibles of \( G \). Let \( O_a \) be an orbit of the previously defined action \( \hat{\phi} \) with representative element \( a \) taken from a complete-set of representatives \( \hat{R} \) of the \( \hat{\phi} \)-partition of \( \hat{A} \)

\[
A = \bigcup_{a \in \hat{R}} O_a \tag{A.4}
\]

Then the following theorem provides a characterisation of a complete set of inequivalent irreducible representations of \( G \).

**Theorem A.1.3** (Little group method). *Given the \( \hat{\phi} \)-partition \([A.6]\) of the character group \( \hat{A} \) defined by a group of automorphisms \( H \cong \langle \varphi_1, \ldots, \varphi_n \rangle \) of \( A \) and a complete-set of irreducible representations of \( H \), then the following extension-rules characterise a complete-set of inequivalent irreducible representations of the semi-direct product \( G = H \ltimes A \).

1. For each \( \chi_a \in \hat{R} \) let \( H_a \subset H \) bet its stabiliser subgroup and \( \rho_a \) an irreducible representation of \( H_a \). Then the representation \( \rho_a \otimes \chi_a \) is an irreducible representation of \( H_a \times A \), subgroup of \( H \ltimes A \), and the induced representation \( \rho_a \otimes \chi_a := \rho_a \otimes \chi_a \uparrow \) is an irreducible representation of \( H \times A \)

2. Two of these irreps are equivalent \( \rho_a \otimes \chi_a \simeq \sigma_a \otimes \chi_a' \) if and only if \( O_a = O'_a \) and \( \rho_a \simeq \sigma_a' \)

3. Every irrep of \( H \ltimes A \) is equivalent to one of the form \( \rho_a \otimes \chi_a \)

**Proof:** Confer Serre’s \([60]\) section 8.2, proof of the proposition 25. Observe that, in practice, it is enough to study the action of the generators \( \varphi_i \) on \( \hat{A} \) to compute stabilisers. ■

### A.2 Application: the irreducible representations of \( \mathbb{Z}_p \ltimes \varphi A \)

Now we apply the method to our case: \( H \) is an abelian group generated by a group of automorphism \( \varphi \) of prime-order \( p \). We identify \( m \leftrightarrow \varphi^m \) and simply write \( H = \mathbb{Z}_p \). This group is cyclic and simple (it has no proper subgroups), therefore, the stabilizer subgroups \( H_a \) can only be trivial \( H_a \simeq \{0\} \) or the entire group \( H_a \simeq \mathbb{Z}_p \). Using the group action defined by the automorphism \( \hat{\varphi} \)

\[
m \cdot (\chi_a) = \chi_{\hat{\varphi}^m(a)} = \chi \circ \varphi^{-m} \tag{A.5}
\]

the orbits of the partition of \( A \) induced by the action \( \hat{\varphi} \) can be of two types.

- Orbits with one element \( O_{a_r} = \{a_r\} \) represented by an element \( a_r = \hat{\varphi}(a_r) \) from the group \( A \).
- Orbits with \( p \)-elements \( O_{b_r} = \{b_r, \varphi(b_r), \ldots, \varphi^{p-1}(b_r)\} \) represented by an element \( b_r \neq \hat{\varphi}(b_r) \) from the group \( A \).
A.2. APPLICATION: THE IRREDUCIBLE REPRESENTATIONS OF $\mathbb{Z}_p \ltimes \varphi A$

Denoting by $\hat{R}_1$ and $\hat{R}_p$ the sets of representatives for each kind or orbit, the partition of $A$ can be written as

$$A = \bigcup_{a_r \in \hat{R}_1} O_{a_r} \bigcup_{b_r \in \hat{R}_p} O_{b_r}$$

(A.6)

where all unions are disjoint. A complete set of irreps of $\mathbb{Z}_p$ can be written in terms of the primitive $p$-th root of the unity $\omega_p := e^{2\pi i/p}$ as $\omega_p^m$ for any $m \in \mathbb{Z}_p$. Then, we can apply the little group method to classify all the irreps of $\mathbb{Z}_p \ltimes \varphi A$, which is the content of the next theorem.

**Theorem A.2.1.** Given a semi-direct product $\mathbb{Z}_p \ltimes \varphi A$ where $A$ is an abelian group, $p$ is prime and $\varphi$ is a group automorphism of $A$ or order $p$, let $\hat{R} = \hat{R}_1 \cup \hat{R}_p$ be a set of representatives of the $\varphi$-partition of the abelian group $A$ defined in eq. (A.6), then the following rules characterise a complete set of inequivalent irreducible representations of $\mathbb{Z}_p \ltimes \varphi A$.

1. **One-dimensional representations.** Any element $a_r$ of the group $A$ fulfilling $a_r = \varphi(a_r)$, representative of an orbit with one element $O_{a_r} = \{a_r\}$, corresponds to a character function $\chi_{a_r}$ that defines, via induction, $p$ inequivalent one-dimensional irreducible representations of $\mathbb{Z}_p \ltimes \varphi A$ with expressions

$$\chi_{m,a_r}(n,b) := (\omega_p^m \cdot \chi_{a_r})(n,b) := \omega_p^m \chi_{a_r}(b)$$

where $m$ takes all values from $0$ to $p-1$. Two irreducible representations $\chi_{m,a_r}$, $\chi_{m',a_r'}$ are equivalent if and only if $(m,a_r) = (m',a_r')$ and any one-dimensional irreducible representation of the group has to be of this form.

2. **High-dimensional representations.** For any other representative element of the group $b_r \in \hat{R}_p$, which fulfils the opposite condition $b_r \neq \varphi(b_r)$, the orbit of $p$-elements represented by the element $O_{b_r} = \{b_r, \varphi(b_r), \ldots, \varphi^{p-1}(b_r)\}$ defines one $p$-dimensional irrep $\sigma_{b_r}$ via induction.

$$\sigma_{a}(n,b) := Z_{O_a}(b)X_{O_a}(n)$$

(A.8)

Where $X_{O_a}(n)$ and $Z_{O_a}(n)$ are defined as

$$X_{O_a}(n) := \sum_{i=0}^{p-1} |i + n\rangle \langle i|$$

$$Z_{O_a}(b) := \sum_{i=0}^{p-1} \chi_{b_r}(a)|i\rangle \langle i|$$

(A.9)

and two irreducible representations $\sigma_{b_r}$, $\sigma_{b_r'}$ are equivalent if and only if $b_r$ and $b_r'$ belong to the same orbit, which guarantees that the definition $\sigma_{b_r}$ is independent of the choice of representative.

**Proof:** We apply theorem A.1.3. Because the stabilisers subgroups of $\mathbb{Z}_p$ are trivial or the entire group (due to simplicity), there can only be orbits with 1 or $p$ elements. We now prove the two cases independently.

**Case 1:** the stabiliser of $a_r = \varphi(a_r)$ is $H_{a_r} = \mathbb{Z}_p$ whose $p$ irreps are the functions $\omega_p^m$. Then extended irreps $\chi_{m,a_r} = \omega_p^m \chi_{a_r}$, for every $m$ from $\mathbb{Z}_p$ are inequivalent irreducible representations of $\mathbb{Z}_p \ltimes \varphi A$.

**Case 2:** for any orbits of $p$ elements $O_{b_r}$ the stabiliser of its representative is trivial $\{0\}$ and its only irreducible is the trivial one. Then the induction

$$\frac{1}{|\hat{R}_p|} \cdot \chi_{b_r} = \chi_{b_r} \\ 1_{\{\{0\}\} \times A}$$

(A.10)

is an irreducible of dimension $p$. Using that the quotient $G/A = \mathbb{Z}_p$ we obtain their unitary representations using

$$\sigma_{a}(n,b) = \sum_{i,j \in \mathbb{Z}_p} \tilde{\chi}((-i,0)(n,b)(j,0)|m\rangle \langle n|$$

(A.11)

Because $\sigma_{a}$ is a homomorphism and $(n,b) = (0,b)[n,0)$, we can decompose

$$\sigma_{a}(n,b) = \sigma_{a}(0,b)\sigma_{a}(n,0)$$

(A.12)
The right term, $\sigma_a(l,0)$ is the $X_P$-Pauli gate, but we denote it $X_{O_a}$ to highlight that the gate is making the irreducibles of $O_a$ "circulate around"

$$
\sigma_a(n,0) = \sum_{i,j} \chi((-i,0)(n,0)+(j,0)) |i\rangle\langle j|
$$
(A.13)

$$
= \sum_{i,j} \delta_{i,j+n} |i\rangle\langle j| = \sum_j |j+n\rangle\langle j|
$$
(A.14)

$$
:= X_{O_a}(n)
$$
(A.15)

The left term $\sigma_a(0,b)$ is a diagonal matrix containing all characters of the orbit $O_a$ and resembles a $Z$-Pauli gate

$$
\sigma_a(0,b) = \sum_{i,j} \chi_a((-i,0)(0,b)) |i\rangle\langle j|
$$
(A.16)

$$
= \sum_{i,j} \chi_a(i-j,\varphi^{-i}(b)) |i\rangle\langle j|
$$
(A.17)

$$
= \sum_i \chi_a \circ \varphi^{-i}(a) |i\rangle\langle i| = \sum_i \chi_{\varphi^i(a)}(b) |i\rangle\langle i|
$$
(A.18)

$$
:= Z_{O_a}(b)
$$
(A.19)

Putting the two factors together we get the correct expressions. Following theorem [A.1.3] step-by-step it follows that the set we have defined is a complete set of inequivalent irreducible irreps.
Appendix B

Block-diagonal formulas

In this section we prove that the unitary gate $F$ given in theorem 6.2.2 block-diagonalises the left and right regular representations ‘exactly’ as we said in the original definition 1.4.1.

$Z_L(n,c) := FX_L(n,c)F^\dagger = \bigoplus_{\sigma \in \hat{G}} I_{d_\sigma} \otimes \sigma(n,c)$

$Z_R(n,c) := FX_R(n,c)F^\dagger = \bigoplus_{\sigma \in \hat{G}} \sigma(n,c)^* \otimes I_{d_\sigma}$

By ‘exactly’ we mean in the decompositions of $Z_L$ and $Z_R$ the irreducible representations are expressed in ‘exactly’ the same basis, which is the one obtained from the little group method and it is defined in box 6.2.

B.1 Left regular representation

**Proposition B.1.1.** The quantum Fourier transform $F$ of the group $\mathbb{Z}_p \ltimes \phi A$ block-diagonalises the left regular representation of the group according equation (6.22) using the set of irreducible representations presented in box 6.2.

**Proof:** Apply the abelian quantum Fourier transform $I_p \otimes F_A$ to the left regular representation

$X_L(n,c) = X(n,c) (1_p \otimes U_\phi)^n = X(n) \otimes X_A(c) U_\phi^n$

Use the properties of the abelian Fourier (equations 1.23, 6.3) to obtain the following unitary gate.

$X_L(n,c)^{FA} = X_p(n) \otimes Z_A(c) U_\phi^n$

$= \sum_{m,a' \in A} X_p(n) \otimes Z_A(c) |m, \phi^n(a')\rangle \langle m, a'|$

$= V_{ar} + V_{br}$

Where the non-unitary matrices $V_{ar}$ and $V_{br}$ are supported on states containing representatives of orbits with 1 and $p$ elements respectively, as defined in box 6.1.

$V_{ar} := \sum_{m} X_p(n) |m\rangle \langle m| \otimes \sum_{a_r} \chi_{a_r}(c)|a_r\rangle \langle a_r|$

$V_{br} := \sum_{m,n,i,b_r,c} \chi_{\phi^n + b_r}(c)|m + n, i + n, b_r\rangle \langle m, i, b_r|$
Now apply $B_D$ (eq 6.19) to each of them. From $V_{a_r}$ one obtains the blocks corresponding to one-dimensional irreducible representations:

\[
V_{a_r}^{B_D} = Z_p(n) \sum_m |m\rangle \langle m| \otimes \sum_{a_r} \chi_{a_r}(c) |a_r\rangle \langle a_r| \\
= \sum_{m,a_r} \omega_p^{mn} \chi_{a_r}(c) |m, a_r\rangle \langle m, a_r| \\
= \bigoplus_{m,a_r} \chi_{m,a_r}(c) 
\]

(B.6)

(B.7)

(B.8)

Respectively, we obtain from $V_{b_r}$ the blocks corresponding to $p$-dimensional representations:

\[
V_{b_r}^{B_D} = \sum_{m,n,i,b_r,c} \chi^{\hat{\phi}^{i+n}(b_r)(c)} |i - m, i + n, b_r\rangle \langle i - m, i, b_r| \\
= \sum_{m'} |m'\rangle \langle m'| \otimes \sum_{n,i,b_r,c} \chi^{\hat{\phi}^{i+n}(b_r)(c)} |i + n, b_r\rangle \langle i, b_r| \\
= \sum_{m'} |m'\rangle \langle m'| \otimes \sigma_{b_r}(n,c) \otimes |b_r\rangle \langle b_r| \\
= \bigoplus_{b_r} I_p \otimes \sigma_{b_r}(n,c) 
\]

(B.9)

(B.10)

(B.11)

(B.12)

Putting both together, one obtains the desired result:

\[
Z_L(n,c) = V_{a_r}^{B_D} + V_{b_r}^{B_D} \\
= \bigoplus_{m,a_r} \chi_{m,a_r}(c) \bigoplus_{b_r} I_p \otimes \sigma_{b_r}(n,c) \\
= \bigoplus_{\sigma \in \hat{G}} I_d \otimes \sigma(n,g) 
\]

(B.13)

(B.14)

(B.15)

\begin{proof}
The proof is similar as the previous one. In this case, however, it helps to split the regular representation into two terms, as we did in section 6.1.4. W.l.o.g it is enough that the two following matrices are block-diagonalised as in equation 6.23.

1. $X_R(1,0) := X_R^\dagger \otimes I_A$

2. $X_R(0,c) := \bigoplus_n X_A(-\varphi^n(a)) = \sum_n |n\rangle \langle n| \otimes X_A(-\varphi^n(c))$
\end{proof}
B.2. RIGHT REGULAR REPRESENTATION

If we apply $F$ directly on the first one we obtain:

$$\mathcal{F}X_R(1,0)F^\dagger = B_D(X_p^\dagger \otimes I_A)B_D^\dagger$$  \hspace{1cm} (B.16)

$$= B_D \left( \sum_{a_r} X_p^\dagger \otimes |a_r\rangle\langle a_r| + \sum_{m,a_r} |m\rangle \otimes |i, b_r\rangle\langle i, b_r| \right) B_D^\dagger$$  \hspace{1cm} (B.17)

$$= \sum_{a_r} Z_p^\dagger \otimes |a_r\rangle\langle a_r| + \sum_{m,a_r} |i - m - 1\rangle\langle i - m| \otimes |i, b_r\rangle\langle i, b_r|$$  \hspace{1cm} (B.18)

$$= \sum_{m,a_r} \hat{\omega}^m \otimes |m, a_r\rangle\langle m, a_r| + \sum_{m',a_r} |m' - 1\rangle\langle m'| \otimes |i, b_r\rangle\langle i, b_r|$$  \hspace{1cm} (B.19)

$$= \bigoplus_{m,a_r} \chi^\ast_{m, a_r}(1,0) \bigoplus_{b_r} \sigma^\ast_{b_r}(1,0) \otimes I_p$$  \hspace{1cm} (B.20)

$$= \bigoplus_{\sigma \in \hat{G}} \sigma^\ast(1,0) \otimes I_{d_0}$$  \hspace{1cm} (B.21)

Now we apply $F$ to the second type of right regular gate

$$\mathcal{F}X_R(0, c)F^\dagger = B_D \left( \sum_n |n\rangle\langle n| \otimes Z_A(-\varphi^n(c)) \right) B_D^\dagger$$  \hspace{1cm} (B.22)

$$= B_D \left( \sum_{n,a_r} \chi^\ast_{n, a_r}(c)|n, a_r\rangle\langle n, a_r| + \sum_{n,i,b_r} \chi^\ast_{n,i}(b_r)(c)|n, i, b_r\rangle\langle n, i, b_r| \right) B_D^\dagger$$  \hspace{1cm} (B.23)

$$= \sum_{n, a_r} \chi^\ast_{n, a_r}(c)|n, a_r\rangle\langle n, a_r| + \sum_{n,i,b_r} \chi^\ast_{n,i}(b_r)(c)|i - n, i, b_r\rangle\langle i - n, i, b_r|$$  \hspace{1cm} (B.24)

$$= \sum_{n, a_r} \chi^\ast_{n,a_r}(0, c)|n, a_r\rangle\langle n, a_r| + \sum_{n,i,b_r} \chi^\ast_{n,i}(b_r)(c)|n', i, b_r\rangle\langle n', i, b_r|$$  \hspace{1cm} (B.25)

$$= \sum_{n, a_r} \chi^\ast_{n,a_r}(0, c)|n, a_r\rangle\langle n, a_r| + \sum_{b_r} Z_{b_r}^\dagger (0, c) \otimes I_p |b_r\rangle\langle b_r|$$  \hspace{1cm} (B.26)

$$= \bigoplus_{n, a_r} \chi^\ast_{n,a_r}(0, c) \bigoplus_{b_r} \sigma^\ast_{b_r}(0, c) \otimes I_p$$  \hspace{1cm} (B.27)

$$= \bigoplus_{\sigma \in \hat{G}} \sigma^\ast(0, c) \otimes I_{d_0}$$  \hspace{1cm} (B.28)
APPENDIX B. BLOCK-DIAGONAL FORMULAS
Appendix C

Groups with small $p$ and the condition ARITH

In this section we show that ‘realistic’ semi-direct products $\mathbb{Z}_p \ltimes \varphi A$ with ‘small’ $p$ always fulfil the condition ARITH (definition 6.2.1). By realistic we mean that multiplication inside the group can be done efficiently classically. By small $p$ we mean $p = O(\text{polylog}|A|)$.

C.1 Small $p$ implies ARITH

Now we prove the statement. We use several results and definitions from chapter 5.

**Lemma C.1.1.** Given a semi-direct product $\mathbb{Z}_p \ltimes \varphi A$ where $A$ is abelian and $\varphi$ is a group automorphism of primer order $p = O(\text{polylog}|A|)$ such that the abelian group $A$ is presented as a direct product of cyclic factors $A = \mathbb{Z}_{N_1} \times \ldots \times \mathbb{Z}_{N_d}$. Then, if the canonical matrix $M_\varphi$ of the automorphism $\varphi$ is given as an input the group $\mathbb{Z}_p \ltimes \varphi A$ fulfils ARITH.

**Proof:** Using $M_\varphi$ we can always compute $\varphi$ as we saw in lemma 5.2.1. Given this fact and that $p$ is small, lemma 5.3.2 shows that we can multiply and exponentiation in the group. Now, if we are able to show how to compute $\hat{\varphi}$ or its inverse we can apply the same kind of reasoning to show conditions 2 and 3 in ARITH. Now, we will show how to compute the canonical matrix of $\varphi^* := \hat{\varphi}^{-1}$ from the matrix $M_\varphi$, from where the rest will follows. Definition 6.1 shows that the matrix $M_{\varphi^*}$ we are looking is the one such that the element $a^* := M_{\varphi^*} \cdot a^T$ fulfils

$$\chi_{\varphi^*(a)}(b) = \chi_{a^*}(b)$$

for all elements $a, b \in A$. Now we show how to compute $M_{\varphi^*}$ constructively.

Remember the definition $[M_\varphi]_{i,j} := \varphi(e_j)$. Using definition 6.1 it follows:

$$\chi_{\varphi^*(a)}(b) := \chi_{a^*}(b)$$ (C.1)

$$= \exp \left\{ 2\pi \sum_i a_i \left( \sum_j [M_\varphi]_{i,j} b_j \right) / N_i \right\}$$ (C.2)

$$= \exp \left\{ 2\pi \sum_j b_j \left( \sum_i [M_\varphi]_{i,j} N_j / N_i a_i \right) / N_j \right\}$$ (C.3)

From here we define the matrix $[M_{\varphi^*}]_{i,j} := [M_\varphi]_{j,i} N_i / N_j$. If this matrix were an integer matrix then it would be the matrix we are looking for. In fact, it is an integer matrix, which can be seen using that the order of the generator $e_i$ is $N_i$ and since $\varphi$ is an automorphism is must hold $N_i \varphi(e_i) = 0$. This
implies \([M_\varphi]_{j,i} N_i = 0 \mod N_j\) for all the components of the tuple, hence \([M_\varphi]_{j,i} N_i = k N_j\) and therefore \([M_\varphi]_{j,i} N_i/N_j = k\) is an integer.

Now, from the matrix \([M_\varphi]\) the quantities \(\hat{\varphi}^m(b) = \hat{\varphi}^{-(p-m)}(b)\) can be done efficiently since \(p = O(\text{polylog}|A|)\) implies that \(p\) consequent matrix multiplications can be done in time \(O(\text{polylog}|A|)\). This is enough to prove conditions 2. and 3. in ARITH, and we are done. \(\blacksquare\)
Appendix D

Engineering a quantum Fourier transform

As we commented in chapter 6, in this work we have not used particular techniques to develop our quantum Fourier transforms. In this appendix we comment how we obtained our formulas. In particular, we prove that the quantum Fourier transform $F_\times$ of the abelian direct-product group $\mathbb{Z}_p \times A$ can be used to implement a Strong Fourier sampling for any semi-direct group $\mathbb{Z}_p \rtimes \varphi A$, generalising a result given in [17] for some wreath-product groups.

However the transform $F_\times$ does not block-diagonalise the regular representations of the group in the form given in definition 1.4.1. Instead, it uses a different choice of basis for the irreps for $\mathbb{Z}_L$ and $\mathbb{Z}_R$. Moreover, in the block-diagonalised gates $\mathbb{Z}_L$ and $\mathbb{Z}_R$ each copy of a high-dim irrep appears, as well, in a slightly different basis. In the circuits we presented in chapter 6 we fixed these problems with appropriate choices of basis for the irreducible representations; in this appendix we show our original calculations.

D.1 Fourier transforms for wreath-products

In [17] the authors found a quantum Fourier Transform for the a family of non-abelian groups known as ‘wreath-products’ $W_n := \mathbb{Z}_2 \rtimes \varphi \mathbb{Z}_2^n$ that allows to solve the Hidden Subgroup Problem for these groups in quantum polynomial time. These groups are semi-direct products of the form $\mathbb{Z}_p \rtimes \varphi A$ where $A = \mathbb{Z}_n^2 \times \mathbb{Z}_n^2$.

The automorphism $\varphi$ is defined as

$$\varphi : \mathbb{Z}_n^2 \times \mathbb{Z}_n^2 \rightarrow \mathbb{Z}_n^2 \times \mathbb{Z}_n^2$$

$$\varphi(a_1, a_2) := (a_2, a_1) \quad \text{(D.1)}$$

Any elements $a$ of this group is a tuple with two $n$-bit strings $a = (a_1, a_2)$. In connection with definition 4.1.1 in [17] the authors use the black operation of the group $\bullet$. Remember that in this thesis we have worked all the time with the white operation $\circ$, hence, the groups defined in [17] are opposite to ours in the sense explained in chapter 4. Now, using the automorphism gates $U_\varphi$ and $CU_\varphi$ defined as

$$U_\varphi := \sum_{a \in A} |\varphi(a)\rangle \langle a| = \sum_{(a_1, a_2)} |a_2, a_1\rangle \langle a_1, a_2| \quad \text{CU}_\varphi |m, a\rangle := |m\rangle U^m_\varphi |a\rangle \quad \text{(D.3)}$$

one of the results of [17] shows that a non-abelian quantum Fourier transforms for these groups can be implemented efficiently with the following quantum circuit:

**Proposition D.1.1.** The matrix $F_n = CU_\varphi \cdot H^{2n+1} \cdot CU_\varphi^\dagger$ implements a Fourier transform of the group $W_n$. The gate is depicted in figure D.1.

In [17] the authors also show that this transform can be used to solve the hidden subgroup problem over $W_n$ using strong Fourier sampling. For details consult the original reference.
APPENDIX D. ENGINEERING A QUANTUM FOURIER TRANSFORM

| A generalisation |

The quantum Fourier transform from figure D.1 is intriguing from a group-theoretical point of view, since it is the abelian quantum Fourier transform of the group $\mathbb{Z}_{2^{n+1}}$ written in a different basis $CU_\phi$. Moreover, the basis $CU_\phi$ is not arbitrary since it implements the group isomorphism that transforms the wreath-product defined with the black operation $\bullet$ into its opposite group, i.e., the group defined with the white operation $\circ$ (definitions 4.1.1, 4.1.2). We prove this fact now.

**Proposition D.2.1.** The unitary gate $CU_\phi$ implements the group isomorphism $\text{Inv}: W_n \rightarrow W_{op}^n$ from definition 4.1.1.

**Proof:** From the definition of the group it follows that the automorphism has order two so $\varphi = \varphi^{-1}$. Also, any element of the group fulfills $(m, a) = (-m, -a)$. Using this two properties one obtains that the inverse of an element is $(m, a)^{-1} = (-m, -\varphi^{-m}(a)) = (m, \varphi^m(a))$ and therefore $U_{\text{Inv}}(m, a) = |m, \varphi^m(a)\rangle = CU_\phi|m, a\rangle$.

From now on we will work in the white group picture. An immediate consequence of this result is that the unitary gate $H^{2n+1}$ is a quantum Fourier transform of the group $W_{op}^n$. This is a very curious result and one might wonder if there is a connection between the quantum Fourier transforms of the groups $\mathbb{Z}_p \rtimes \varphi A$ and $\mathbb{Z}_p \times A$ in general.

In fact, we show that this is the case: for any group $\mathbb{Z}_p \rtimes \varphi A$ a quantum Fourier Transform can be written as $F = PDF_\times$, where $P$ is a term implementing the re-labelling we saw in box 6.3, $D$ is a diagonal unitary that defines a choice of basis for the irreducible representations of the group (box 6.2) and $F_\times$ is the Fourier Transform of the abelian group $\mathbb{Z}_p \times A$. We prove this result constructively showing that there is a unitary gate with the form $PDF_\times$ that block-diagonalises the left-regular representation of the group (section 6.1.4). A consequence of Schur’s lemma is that a gate fulfilling this property must block-diagonalise the right-regular representation as well [25].

**Theorem D.2.2.** For any semi-direct product of the form $\mathbb{Z}_p \rtimes \varphi A$ where $A$ is an abelian group and $\varphi$ is a group automorphism of primer order $p$ there is a quantum Fourier transform that can be written in the following form

$$F := PDF_\times$$

where $P$ is a permutation implementing the re-labelling from box 6.3 and $D$ is a diagonal gate that sets a basis for the high-dimensional irreducible representations (box 6.2).

**Proof:** Since the wreath-products fulfil ARITH (cf. appendix C) the term $P$ is not very important since it is just an efficient classical operation which changes the labels of the final computational basis. With this in mind, we prove the theorem in the following 2 steps.

1. We apply $F_\times$ to the left-regular-representation of $\mathbb{Z}_p \rtimes \varphi A$ and show that from the resulting equivalent-representation all the irreps that should appear in the decomposition of the left-regular

---

Note: unlike in definition 1.4, a Fourier transform constructed in this way may block-diagonalise each regular representation $X_L, X_R$ using a different choice of basis for the irreps.
representation of $\mathbb{Z}_p \ltimes \sigma A$ can be extracted using a complete-set of orthogonal projectors. More rigorously, for each $m$ copy of each high-dimensional irrep $\sigma$ we define a unique projector $P_{m,\sigma}$ such that the projected matrix

$$P_{m,\sigma} \left( F_\chi X_L(n, c) F_\chi^\dagger \right) P_{m,\sigma} \simeq \sigma_m \quad \text{and} \quad \sum_{\sigma \in \mathbb{Z}_p} \sum_{m=0}^{d_{m,\sigma} - 1} P_{m,\sigma} = I_{W_n^{\sigma}}$$

is equivalent to the $m$-th copy of the irrep $\sigma$ that has to appear in the Fourier transformed regular representation. Therefore $F_\chi$ block-diagonalises the regular representation $X_L$. However, we also show that multiple copies of the same irrep are expressed in different bases in this decomposition.

2. We add a correcting terms $D$ to write all copies of the same irrep in the same basis.

**Step 1.** Use that $F_\chi = F_p \otimes F_A$ where the right term is $F_A = \sum_{a,b \in A} \chi_A(b)|a\rangle\langle b|$, and $F_p = \sum_{i, j \in \mathbb{Z}_p} \omega_{ij}^n |i\rangle\langle j|$. Apply it to the left regular representation of $\mathbb{Z}_p \ltimes \sigma A$ with matrix

$$X_L(n, c) = X(n, c) (I_p \otimes U_\sigma)^n$$

We transform both matrices in this product separately. The first term, the $X(n, b)$ gate, turns by definition into

$$F_\chi X(n, c) F_\chi^\dagger = Z(n, c)$$

To compute the second term we use that $F_\chi (I_p \otimes U_\sigma) F_\chi^\dagger = I_p \otimes (F_A U_\sigma F_A^\dagger)$ and that Fourier-transforming the automorphism gate $U_\sigma$ yields the dual automorphism gate $U_{\bar{\sigma}}$. Put everything together to get

$$Z_L(n, c) = Z(n, c) \cdot (I_p \otimes U_\sigma)^n$$

Now we show that this gate contains all irreducibles from box 6.2 using a following complete-set of orthogonal projectors. For each one dimensional irrep $\chi_{m,a_r}$ define the projector $P_{m,a_r} := |m, a_r\rangle\langle m, a_r|$. We project out an irrep using the change of basis $|m\rangle\langle m| \otimes Z^m p$. To see that the projected-out matrix is equivalent to $\sigma_{b_r}$ use the change of basis $|m\rangle\langle m| \otimes Z^m p$ where $Z^m p = \sum_{i=0}^{p-1} \omega^{mp} \cdot |i\rangle\langle i|$. 

**Step 2** A consequence of the previous part is that in the block-diagonalised regular-representation gate $Z_L$ equivalent copies of $p$-dimensional irreps $\sigma_{n,c}$ differ by a phase $\omega^{mn}$ which is added or removed with $Z_p^m$-Pauli gates. We can remove this phases adding the following correcting term

$$D := \sum_{m, a_r} |m, a_r\rangle\langle m, a_r| + \sum_{m, b_r, i} \omega^{mi} |m, \tilde{\phi}(b_r)\rangle\langle m, \tilde{\phi}(b_r)|$$

Finally, at the end we add the re-labelling $P$ to present the block-diagonalised unitaries in the re-labelled basis $P$. This ends the proof.
The previous theorem does not prove that $H^{2n+1}$ is a Fourier transform but it does show that this gate can be used as a QFT for the wreath-products if we perform strong Fourier sampling (like in [17]). The reason is the following, if we let the quantum Fourier transform of the theorem $F$ act a coset-state and we then perform a final projective measurement w.l.o.g. the term $P$ can be included in the classical post-processing\(^2\). Moreover, the diagonal term $D$ does not affect the probability distribution of the outcomes if we do a measurement in the computational basis. Therefore, the part of the transform $F$ relevant in strong Fourier sampling is $F_x$. In the case of the wreath-products, $F_x = H^{2n+1}$.

Finally, in connection with the quantum Fourier transforms of chapter 6, we can implement the transform $F$ as the following quantum circuit. Which is a quantum Fourier transform of the form given in theorem 6.4.1 and from the proof of this theorem it holds that the quantum Fourier transform $F$ is not canonical.

\[ |m\rangle \quad F_P \quad F_P \quad Z_P \quad |a\rangle \]

\[ F_A \]

---

**Remark.** Since the re-labelling $P$ is just a classically computable permutation for groups fulfilling ARITH, we often do not consider this term explicitly in chapters 6 and 7.

---

\(^2\)Realise that the wreath-products fulfil ARITH (cf. appendix C).
Bibliography


[58] “Computational complexity of mathematical operations,” in *Wikipedia*.


Acknowledgements

First of all, I want to thank Prof. Dr. Ignacio Cirac for being my diploma thesis supervisor and for giving me the opportunity to work in such a fascinating academic environment as the MPQ Theory division.

I want to thank my second supervisor Dr. Maarten Van den Nest for providing a highly interesting topic for my thesis, his constant help, his patience with my innumerable questions during this year and for several suggestions on this manuscript.

I am grateful to Prof. Dr. Andrew Childs from University of Waterloo for pointing out useful references.

I give special thanks to all my colleagues and friends from the MPQ Theory division, who all contributed to create a pleasant and friendly working atmosphere and because of many good seminars and workshops. I am very grateful to Dr. Miguel Aguado and Dr. Oliver Buerschaper for helpful academic discussions; to my officemates Dr. Christine Muschik, Dominik Kohler, Xiaotong Ni, Thorsten Wahl and Alexander Müller-Hermes for their companion during the year; and to Veronika Lechner for countless hours of administrative help.

To my family, in special to my parents and brother, I give my deepest thank for their love and support during my education.

Gracias Esther por tu apoyo los últimos meses.

This project was supported jointly by “La Caixa” and “Deutscher Akademischer Austauschdienst” (DAAD).