Multidimensional Quantum Walks and the Multiplicative Ladder Adversary

Sebastian Zur

Multidimensional Quantum Walks and the Multiplicative Ladder Adversary

Sebastian Zur

Multidimensional Quantum Walks and the Multiplicative Ladder Adversary

Sebastian Arkadiusz Zur

This research was supported by the Dutch Research Council (NWO) through the grant *Quantum time-space tradeoff lower bounds* (OCENW.Klein.061).

Cover design by Chelsea Wegman.
Printed and bound by Ipskamp Printing.

ISBN: 978-94-6473-773-8

Multidimensional Quantum Walks and the Multiplicative Ladder Adversary

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde commissie,
in het openbaar te verdedigen in de Aula der Universiteit
op woensdag 30 april 2025, te 11.00 uur

door Sebastian Arkadiusz Zur
geboren te Utrecht

Promotiecommissie

Promotor:	prof. dr. S.M. Jeffery	Universiteit van Amsterdam
Copromotor:	dr. K. Guo	Universiteit van Amsterdam
Overige leden:	prof. dr. R.M. de Wolf	Universiteit van Amsterdam
	prof. dr. J.A. Ellis-Monaghan	Universiteit van Amsterdam
	dr. M. Ozols	Universiteit van Amsterdam
	prof. dr. M. Walter	Ruhr-Universität Bochum
	dr. A. Belovs	University of Latvia

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

Contents

II Multiplicative Ladder Adversaries

Chapter 1

Introduction

Als je het opschrijft, staat het meteen op papier ook.

Gerard Reve

The rise of computation has profoundly transformed the way we solve problems. From the invention of early mechanical calculators to the advent of modern supercomputers, the ability to perform large-scale calculations has empowered us to tackle increasingly complex tasks. Today, computers play a critical role in nearly every aspect of society, from simulating physical phenomena to performing data analysis on an unprecedented scale. This computational revolution is built on a foundation of well-designed *algorithms*—explicit, step-by-step instructions that guide a computer to solve specific problems efficiently.

Designing algorithms often feels intuitive because humans have relied on algorithmic thinking for centuries in daily life. A cooking recipe, for example, serves as a perfect illustration of an algorithm: it provides clear instructions on which ingredients to use, when to combine them, and how long to cook them. Recipes guide us step-by-step toward a desired outcome, mirroring the structured problem-solving approach of algorithms. Many foundational programming concepts find natural parallels in cooking: conditional actions resemble decision-making in recipes ("*If the sauce is too acidic, add sugar*"), iterative processes mimic repeated tasks ("*Stir continuously until the sauce emulsifies*"), and parallelism corresponds to multitasking ("*While the aubergine roasts in the oven, prepare the sauce*"). These analogies highlight how classical algorithms align with familiar, concrete reasoning, allowing us to design, understand, and adapt them with ease.

Unlike *classical computers*, which operate under the laws of classical mechanics, *quantum computers* leverage the principles of quantum mechanics, such as superposition and entanglement. These principles sometimes enable *quantum algorithms* to outperform their classical counterparts, for instance, in factoring large numbers [Sho97], searching unstructured databases [Gro96], and simulating quantum systems [Llo96]. In the field of *quantum computation*, researchers investigate the power and limitations of quantum algorithms, seeking to better understand which problems admit significant speedups and which do not.

Despite these remarkable advantages, designing quantum algorithms presents unique challenges. While classical algorithms often follow intuitive patterns, quantum algorithms demand a fundamentally different mindset. Crafting a quantum algorithm is less like following a recipe and more akin to composing a musical piece. Quantum computers rely on interference patterns to amplify correct solutions and cancel out incorrect ones. This process resembles creating harmony in music: constructive interference acts like a

well-tuned chord, while destructive interference eliminates dissonance. However, even a slight misalignment in the algorithm's design—caused by poorly chosen gates or incorrect timing—can disrupt this delicate balance, rendering the computation ineffective. Just as a musical harmony that is slightly out of tune sounds jarring, a quantum algorithm that fails to align perfectly with the problem's structure becomes unproductive.

This lack of intuitiveness arises from the unique principles underlying quantum mechanics. Unlike classical operations, quantum operations must be reversible; information cannot simply be copied or deleted. Moreover, the outcomes of quantum computations are inherently probabilistic. These challenges mean that designing quantum algorithms requires not only experience in the discrete branches of mathematics and theoretical computer science, as is the case for classical algorithms, but also a deep understanding of quantum mechanics.

In this thesis, we explore frameworks designed to make studying the capabilities and limitations of quantum algorithms more accessible. These frameworks, in some sense, abstract away much of the inherent quantum mechanical complexity, reducing problems to a form of "classical difficulty." Remarkably, we show that this reduction still preserves much of the inherent power of quantum algorithms, allowing us to better understand their potential and limitations.

1.1 Designing more powerful quantum walks

In Part I, the first part of this thesis, we explore a class of strategies for designing quantum algorithms. These strategies are known as *quantum walk search frameworks*, and the resulting quantum algorithms are referred to as *quantum walks*. These frameworks often begin with a classical random walk, which may be explicit or implicit in the design.

A random walk is a (Markovian) process that describes a path consisting of a sequence of probabilistic steps, where each step depends only on the current state and is independent of previous steps. A simple example is the "drunkard's walk": imagine a person attempting to return home by taking random directions at each street intersection. Over time, this random motion causes the person's position to spread across the city, exhibiting statistical properties that can be analysed to determine, for instance, the expected time to return home. Random walks are fundamental in probability theory and form the basis for numerous classical algorithms, particularly in searching and optimisation tasks. Their well-understood properties make them an excellent starting point for extending these ideas into the quantum domain.

One such quantum walk search framework, originally introduced by [Bel13], is known as the *electric network framework*. In this approach, the construction and cost analysis of the quantum walk algorithm involve analysing the underlying graph, where the random walk takes place, as if it were an *electrical network*. In this analogy, the weighted edges mimic electrical wires, with the weights representing the conductance of the edges. This framework establishes a fascinating connection between quantum walks and electrical networks, a relation that was further strengthened by [AP22]. While this framework is particularly effective for designing quantum algorithms—even for those with limited knowledge of quantum computing—it has a significant limitation shared with other quantum walk search frameworks: it can provide at most a quadratic speedup over the underlying classical random walk.

In this thesis, we address the question of whether we can strengthen quantum walks based on electrical networks, potentially achieving exponential speedups while maintaining the original intuition. Specifically, we aim to ensure that the construction and cost analysis still rely on random walks and electrical networks. We answer this question in the affirmative.

First, in Chapter 4, we enhance these quantum walks using the *multidimensional quantum walk framework*. This approach introduces a new class of quantum walks capable of achieving exponential speedups. However, in doing so, we partially break the connection with electrical networks. To restore this relationship, we generalise electrical networks in Chapter 6. These generalisations allow us to re-establish the connection between quantum walks and electrical networks while preserving the enhanced capabilities of multidimensional quantum walks. This interplay between quantum walks, electrical networks, and their generalisations is illustrated in Figure 1.1.

To evaluate the quality of an algorithm, we are primarily interested in its *cost*. Various metrics can be used to measure this cost, but here, we focus on the runtime of the algorithm. By considering the optimal algorithm for a specific computational problem—the one with the minimal runtime—we define the cost as the *time complexity* of the problem. In the field of *quantum complexity theory*, we study the cost of computational problems for the optimal quantum algorithm, which is referred to as their *quantum (computational) complexity*.

The main application of our multidimensional quantum walk framework is a new quantum walk for the problem of k*-distinctness*. In this problem, we are given access to a list of n integers, and we must determine whether the list contains k copies of the same integer. This problem has been studied as a fundamental problem in query complexity, as it serves as an important subroutine for other more practical problems. Classically, any algorithm solving this problem with bounded error must query a constant fraction of the entire list. Quantum algorithms, however, can achieve significantly better performance. The time complexity of our new quantum walk for k-distinctness matches the best-known query upper bound up to polylogarithmic factors [Bel12a].

Intuitively, it seems reasonable to expect that the time complexity of k-distinctness should not be significantly higher than its query complexity—after all, what else could an algorithm do beyond querying and comparing inputs? However, this upper bound had not been proven through the explicit construction of a quantum algorithm, making it infeasible to analyse its time complexity directly. Our framework bridges this gap by explicitly constructing such an algorithm and providing insights into its runtime performance.

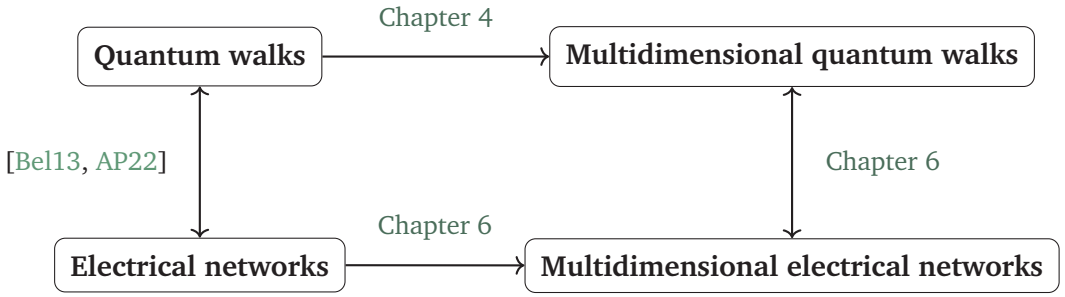


Figure 1.1: Connections between and generalisations of quantum walks and electrical networks

1.2 Limitations of quantum algorithms

While designing new and faster quantum algorithms to tackle computational problems is fascinating and helps reduce the quantum complexity of these problems, it is equally important to understand their limitations. The most valuable component in quantum algorithm research is arguably not the availability of low-fidelity gates or error-corrected qubits, but a researcher's time. By proving lower bounds on the quantum complexity of a problem, we demonstrate that no quantum algorithm can outperform this lower bound.

While perhaps less exciting, these no-go results play a crucial role in guiding the field by clarifying which speed-ups are (un)achievable. In practice, however, it is exceedingly difficult to lower-bound the quantum complexity of computational problems when considering the optimal runtime. Instead, a simpler cost metric is considered, namely the number of *queries*, which we elaborate on in the next chapter.

Providing query lower bounds for classical algorithms can often be done intuitively, even if it does not guarantee a tight bound—a lower bound that closely approximates the optimal query cost. For example, if one wanted to estimate the time needed to understand this thesis, they could first lower bound the time required to comprehend a single page and then multiply this by the number of pages. This approach would yield a straightforward, albeit loose, lower bound, as not all pages require the same effort to understand—some might take significantly longer than the "easiest page".

This reasoning, however, cannot be directly applied to the quantum setting. A quantum reader might "read" multiple pages in superposition or entangle the pages containing a theorem with those discussing its applications, potentially enabling a faster way to understand the work. Similarly to constructing quantum algorithms, the quantum setting trades off power for intuitiveness when deriving lower bounds. This trade-off is also evident in the spectrum of techniques used to derive *quantum query lower bounds*. These techniques range from the original *adversary method* by [Amb00] to the *multiplicative adversary method* [Špa08]. While stronger techniques can provide tighter lower bounds over broader parameter regimes, they are generally more mathematically complex and less intuitive to derive.

In Part II, we study quantum query lower bound frameworks, complementing the exploration of quantum (query) upper bounds achieved through the quantum algorithms discussed in Part I. More specifically, we investigate a quantum query lower bound technique known as the *compressed oracle technique* [Zha19]. This method stands out as an exception to the aforementioned trade-off: it derives strong quantum query lower bounds in an intuitive manner but is constrained to a very limited regime of parameters. We demonstrate how this technique can be unified within the framework of the multiplicative adversary method through a novel approach that we introduce as the *multiplicative ladder adversary method*. This new method is a simplification of the multiplicative adversary method, enabling a more intuitive rederivation of the lower bounds capturing the core power of the compressed oracle technique. This unification provides several valuable insights. On the one hand, it offers a more accessible way to derive lower bounds using the multiplicative adversary method. On the other hand, it suggests how the compressed oracle technique could be extended to accommodate a broader range of parameters.

1.3 Organisation

The thesis is structured as follows. In Chapter 2, we present the essential background information, definitions, and notation used throughout this thesis. Additionally, we introduce fundamental concepts of quantum computing that are necessary to understand the majority of the thesis.

The remainder of the thesis is divided into two parts. In Part I, we address the generalisation of quantum walks constructed by the electrical network framework and explore how this leads to a generalisation of electrical networks themselves. Below, we describe the chapters comprising Part I in more detail.

In Chapter 3, we review fundamental graph-theoretic concepts and the basics of electrical networks. We then delve into the construction of quantum walk algorithms using the electrical network framework and see how the cost analysis of these algorithms is intrinsically connected to the properties of electrical networks. Finally, we demonstrate how

these connections facilitate the design of quantum algorithms capable of locating marked vertices in a graph or sampling from the electrical flow within an electrical network.

In Chapter 4, we generalise the electrical network framework to develop the multidimensional quantum walk framework. After constructing this new framework, we apply it to the *welded trees* problem. This problem involves an exponentially large graph consisting of two full binary trees with 2^n leaves, where the leaves of the two trees are connected by a pair of random matchings. The goal is to start at the root of one binary tree and locate the root of the other binary tree. In the welded trees graph, all vertices except s and t have degree 3, while s and t each have degree 2. The oracle to this graph is designed such that any classical algorithm attempting to solve this problem is forced to perform a random walk, starting from s and continuing until it reaches t. Due to the structure of the graph, the walker is quickly drawn towards the centre, but escaping the centre to locate t takes exponential time. This intuition can be rigorously formalised, showing that the classical query complexity of this problem is $2^{\Omega(n)}$ [CCD+03]. Using the novel concept of *alternative neighbourhoods*, our quantum walk traverses the welded trees graph in just $O(n)$ queries and $O(n^2)$ time. This result demonstrates that the multidimensional quantum walk framework is capable of achieving exponential speedups, making it the first quantum walk search framework to do so.

In Chapter 5, we apply our multidimensional quantum walk framework to the problem of k*-distinctness*, resolving a more than 10-year-old open problem regarding a missing time upper bound that matches the best-known query upper bound. In our algorithm, we apply the second novel technique within our framework, called *edge composition*. This technique allows a quantum walk to account for varying traversal costs across edges, replacing the uniform maximum cost assumption with a more efficient cost analysis that incorporates individual edge costs. To build intuition for the general case, we first address the simpler problem of 3-distinctness, before generalising our approach to the k-distinctness case and demonstrating that our quantum walk achieves a time complexity of $\widetilde{O}(n^{\frac{3}{4}-\frac{1}{4}\frac{1}{2^k-1}})$, matching the best-known query upper bound up to polylogarithmic factors.

In Chapter 6, we restore the relationship between quantum walks and electrical networks, which was disrupted by the generalisation introduced through the multidimensional quantum walk framework. Using the concept of alternative neighbourhoods, we derive generalised notions of electrical flow and potential in electrical networks, which we term the *alternative electrical flow* and *alternative potential*. This new multidimensional electrical network framework enables quantum algorithms to generate quantum states corresponding to the alternative electrical flow over the edges of graphs. Additionally, we demonstrate that the alternative flow and potential satisfy the same linear relationships as the standard flow and potential in traditional electrical networks. We then apply this framework to locate a marked vertex in one-dimensional random hierarchical graphs [BLH23]. Finally, we define a family of graphs where the quantum alternative electrical flow state can be efficiently generated and sampled to recover an s-t path exponentially faster than is possible classically.

In Part II, which consists of Chapter 7, we compare different techniques for lower bounding the quantum query complexities of computational problems. Specifically, we examine how the compressed oracle technique compares to the more established multiplicative adversary method. We provide an explicit reduction from the compressed oracle technique to the multiplicative adversary method [Špa08] by introducing a simplified variant of the latter, which we call the *multiplicative ladder adversary (MLA) method*. This simplification allows for more accessible derivations of lower bounds while capturing the compressed oracle technique as well as a recent generalisation by [Ros21]. Consequently, the MLA method positions itself as a promising tool for deriving lower bounds that are both more accessible than those obtained with the multiplicative adversary method and

applicable to a broader range of parameters than the compressed oracle technique.

1.4 List of publications

This thesis is based on the following papers (in chronological order). In each work, all authors contributed equally.

[JZ23] ***Multidimensional Quantum Walks, with Application to k-Distinctness***
Stacey Jeffery, Sebastian Zur,
Proceedings of the 55th Annual ACM Symposium on Theory of Computing (2023),
TheoretiCS, Volume 4 (2025).
Chapter 3, 4 and 5 are based on this paper.

[LZ23] ***Multidimensional Electrical Networks and their Application to Exponential Speedups for Graph Problems***
Jianqiang Li, Sebastian Zur,
arXiv preprint, accepted for publication in Quantum
Chapter 3 and 6 are based on this paper.

[JZ] ***The Compressed Oracle is a Worthy Adversary***
Stacey Jeffery, Sebastian Zur,
In preparation.
Chapter 7 is based on this paper.

CHAPTER 2

Preliminaries

We zijn voorbij de start, maar we zijn nog niet aan het einde van het begin.

Mark Rutte

In this chapter, we present the essential background information, definitions, and notation used throughout this thesis, assuming a basic understanding of linear algebra. We delve into fundamental concepts of quantum computing that are necessary to understand the majority of the thesis. While a comprehensive background in quantum mechanics is not required for the topics we cover, interested readers may refer to [NC00] for an accessible introduction to quantum mechanics in the context of quantum computation. Any concepts that are specific to certain sections or chapters will be introduced at the beginning of those respective sections.

2.1 Notation

We begin by outlining the notational conventions used in this thesis. Any notation specific to particular sections will be introduced when it first appears. For any positive integer $n \in \mathbb{N}_{\geq 1}$, we denote $[n] = \{1, \ldots, n\}$ and $[n]_0 = \{0, \ldots, n\}$. For any finite set Ω, we write 2^Ω for the power set of Ω. For any two finite sets Ω, Γ, we write Γ^Ω for the set of all functions from Ω to Γ. The *Kronecker delta* of any two mathematical objects a, b is defined as

$$\delta_{a,b} := \begin{cases} 1 \text{ if } a = b, \\ 0 \text{ otherwise.} \end{cases} \tag{2.1}$$

2.1.1 Computer science

We take all logarithms to be base 2. The *natural logarithm*, i.e. the logarithm base e, is denoted by $\ln$. A *bit* is a binary digit, taking a value of either 0 or 1. In this context, we consider bits as elements of $\mathbb{F}_2$, where addition is defined modulo 2. This is functionally identical to the XOR operation, denoted by $\oplus$. The set of all n-bit strings is written as $\{0,1\}^n$. For any n-bit string $x \in \{0,1\}^n$, we use x_i to denote its ith bit. For two n-bit strings $x, y \in \{0,1\}^n$, we write $x \oplus y \in \{0,1\}^n$ for their element-wise XOR and $x \cdot y \in \mathbb{N}$ for their inner product, defined as $\sum_{i=1}^n x_i \cdot y_i$.

To analyse the asymptotic behaviour of any function $f : \mathbb{R} \to \mathbb{R}$, we make use of *big O* notation, defined as

$$f(x) = O(g(x)) \iff \exists C, x_0 \in \mathbb{R}_{>0} \text{ such that } \forall x > x_0 : |f(x)| \leq C\,|g(x)|\,.$$

We also rely on the following related notations, known as *small O*, *big Omega* and *big Theta*, respectively:

$$\begin{aligned} f(x) = o(g(x)) &\iff \exists x_0 \in \mathbb{R}_{>0},\ \forall c \in \mathbb{R}_{>0}, \text{ such that } \forall x > x_0 : |f(x)| \leq c\,|g(x)|\,, \\ f(x) = \Omega(g(x)) &\iff g(x) = O(f(x)), \\ f(x) = \Theta(g(x)) &\iff f(x) = O(g(x)) \text{ and } f(x) = \Omega(g(x)). \end{aligned}$$

In addition, we employ the *soft big O* and *soft big Theta* notation, denoted by $\widetilde{O}$ and $\widetilde{\Theta}$, respectively, which suppress polylogarithmic factors in asymptotic bounds:

$$\begin{aligned} f(x) = \tilde{O}(g(x)) &\iff \exists k \in \mathbb{N}_{\geq 1}, \text{ such that } f(x) = O(g(x)\log^k(g(x))), \\ f(x) = \tilde{\Theta}(g(x)) &\iff f(x) = \tilde{O}(g(x)) \text{ and } g(x) = \tilde{O}(f(x)). \end{aligned}$$

When dealing with multiple variables, it is assumed that the big O bound applies independently to each variable in its respective asymptotic regime. For example,

$$f(x, \epsilon) = O\left(\frac{x^2}{\epsilon}\right) \iff \exists C, x_0 \in \mathbb{R}_{>0} \text{ such that } \forall x_1, x_2 > x_0 : |f(x_1, 1/x_2)| \leq C \left|\frac{x_1^2}{1/x_2}\right|.$$

As a variant of big O (and related) notation, we may also use the following to describe the asymptotic growth of functions:

$$\begin{aligned} f(x) \in \mathsf{poly}(x) &\iff \exists k \in \mathbb{N}_{\geq 1}, \text{ such that } f(x) = O(x^k), \\ f(x) \in \mathsf{polylog}(x) &\iff \exists k \in \mathbb{N}_{\geq 1}, \text{ such that } f(x) = O(g(x)\log^k(g(x))). \end{aligned}$$

2.1.2 Linear algebra

In this work, we work with *Hilbert spaces*, which in the finite-dimensional case (the only case considered in this thesis) are vector spaces equipped with an inner product, denoted by $\langle \cdot, \cdot \rangle$. While Hilbert spaces can be defined over both the field of real numbers $\mathbb{R}$ and the field of complex numbers $\mathbb{C}$, in this thesis we assume they are over $\mathbb{C}$, unless stated otherwise. We write I for the identity matrix on any (finite-dimensional) Hilbert space.

For any real $p \geq 1$, including the limit $p = \infty$, the p-norm of any vector $v = (v_1, \ldots, v_d)$ is given by

$$\|v\|_p := \left(\sum_{i=1}^{n} |v_i|^p \right)^{1/p}.$$

On Hilbert spaces, we use the *Euclidean norm* (or *2-norm*), which is induced by the inner product and denoted by $\|v\| := \|v\|_2 = \sqrt{\langle v, v \rangle}$.

For any linear operator $A \in \mathbb{C}^{m \times n}$ between Hilbert spaces $\mathbb{C}^n$ and $\mathbb{C}^m$, we follow the convention in quantum mechanics and denote the *conjugate transpose* of A by $A^\dagger$ and we say that an operator is *Hermitian* if it equals its conjugate transpose. This notation naturally extends to vectors. The notation $A^\dagger$ should not be confused with A^+, which denotes the *Moore-Penrose* inverse, also known as the pseudoinverse, of A. This is the unique linear operator satisfying the following:

$$AA^+A = A, \quad A^+AA^+ = A^+, \qquad (AA^+)^\dagger = AA^+, \qquad (A^+A)^\dagger = A^+A. \tag{2.2}$$

A Hermitian operator A is *positive semidefinite*, denoted by $A \succeq 0$, if and only if $x^\dagger Ax \geq 0$, or equivalently, if all of its eigenvalues are non-negative. Similarly, we say that A is *positive definite*, denoted by $A \succ 0$, if all its eigenvalues are positive. For any two Hermitian operators A, B, we write $A \succeq B$ if their difference $A - B$ is positive semidefinite, and we write $A \succ B$ is their difference is positive definite.

Similarly as we did with vectors, for any real $p \geq 1$, including the limit $p = \infty$, its p-norm is defined as

$$\|A\|_p := \sup_{v \neq 0} \frac{\|Av\|_p}{\|v\|_p}. \tag{2.3}$$

For $p = 1$ and $p = \infty$, these are simply the maximum absolute column sum and row sum of the matrix A, respectively. In this thesis, the norm of any linear operator A refers to its *spectral norm*, defined as follows:

$$\|A\| := \|A\|_2 = \sqrt{\lambda_{\max}(A^\dagger A)}.$$

We will frequently need to bound the spectral norm in terms of simpler norms, and for this purpose, we use the following lemma, which is a standard result in linear algebra:

Lemma 2.1.1. *For any linear operator A, the spectral norm of A satisfies*

$$\|A\| \leq \sqrt{\|A\|_1 \|A\|_\infty}.$$

2.1.3 Dirac notation

Throughout this work, we employ *Dirac notation*. A *ket*, written as $|\cdot\rangle$, denotes a vector in a Hilbert space, typically of norm 1 unless stated otherwise. If a Hilbert space $\mathcal{H} = \mathbb{C}^\Omega$ is associated with a finite set Ω, its standard basis vectors are denoted by $|\omega\rangle$ for $\omega \in \Omega$. We may use the shorthand $\mathbb{C}[\Omega] = \mathbb{C}^\Omega$ when convenient. For any $d \in \mathbb{N}_{\geq 1}$ we denote by $\mathbb{C}^d$ the Hilbert space $\mathbb{C}^{[d-1]_0}$ of dimension d, with its standard basis given by the set

$\{|i\rangle : i \in [d-1]_0\}$. We refer to this standard basis on $\mathbb{C}^d$ as the *computational basis*. As these vectors correspond to states of some implicit quantum system, we will refer to them as *states* instead of vectors.

A *bra*, written as $\langle\cdot|$, represents a vector in the dual of a Hilbert space and acts as a linear form on vectors by taking the inner product. Given a vector $|v\rangle$ in $\mathcal{H}$ and a vector $|u\rangle$ in the dual of $\mathcal{H}$, the linear form $\langle u|$ acts on $|v\rangle$ by taking the inner product between $|v\rangle$ and $|u\rangle$, denoted by $\langle u|v\rangle$. Since each ket represents a quantum state, it is the convention in quantum mechanics to assume the inner product to be linear in the second variable (the ket) and antilinear in the first (the bra).

If $\{|u_1\rangle, \ldots, |u_n\rangle\}$ and $\{|v_1\rangle, \ldots, |v_m\rangle\}$ are orthonormal bases for two Hilbert spaces $\mathcal{H}_u$ and $\mathcal{H}_v$ of dimension n and m, respectively, then their *tensor product space* is the nm-dimensional space $\mathcal{H}_u \otimes \mathcal{H}_v$ spanned by the orthonormal basis $\{|u_i\rangle \otimes |v_j\rangle : i \in [n], j \in [m]\}$, where $\otimes$ denotes the *Kronecker product* between two matrices. Often we omit the tensor product symbol for such vectors, abbreviating $|u_i\rangle \otimes |v_j\rangle$ to $|u_i\rangle|v_j\rangle$ or even $|u_i, v_j\rangle$.

2.2 Quantum computation

In this section, we introduce fundamental concepts of quantum computing that will be useful throughout the remainder of this thesis. This overview is intended as a brief summary; for a more comprehensive treatment of the subject, we refer the interested reader to [dW19].

2.2.1 Qubits

The fundamental unit of information in quantum computing is the quantum bit, or *qubit*. A qubit represents a quantum state in the two-dimensional Hilbert space $\mathbb{C}^2 = \operatorname{span}\{|0\rangle, |1\rangle\}$.

According to the principles of quantum mechanics, the state space of a composite physical system is described by the tensor product of the state spaces of its individual subsystems. Specifically, for an n-qubit system, the state space is $(\mathbb{C}^2)^{\otimes n}$, spanned by the set $\{|i_1\rangle \otimes \cdots \otimes |i_n\rangle : i_1, \ldots, i_n \in \{0, 1\}\}$.

For convenience, the Hilbert space $(\mathbb{C}^2)^{\otimes n}$ is often thought of as isomorphic to $\mathbb{C}^{2^n}$, simplifying the representation of multi-qubit states. Specifically, for any n-bit integer $i \in [2^n - 1]_0$ with binary representation $i = i_1 i_2 \cdots i_n$, the corresponding computational basis state in $\mathbb{C}^{2^n}$ is written as $|i\rangle = |i_1\rangle \otimes \cdots \otimes |i_n\rangle$. It is important to note that not every quantum state can be decomposed into the product of 1-qubit states. Such states are referred to as *entangled*, and they play a key role in quantum mechanics, distinguishing it from classical physics. A well-known example of an entangled state is the Einstein-Podolsky-Rosen (EPR) state [EPR35], which resides in $(\mathbb{C}^2)^{\otimes 2}$:

$$|\mathsf{EPR}\rangle := \frac{1}{\sqrt{2}} \left(|00\rangle + |11\rangle\right). \tag{2.4}$$

Since any n-qubit state $|\psi\rangle$ is a normalised vector in $\mathbb{C}^{2^n}$, it can be expressed as

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle,$$

where $\alpha_1, \ldots, \alpha_{2^n-1} \in \mathbb{C}$ and $\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$. In this representation, we say that $|\psi\rangle$ is in a *superposition* over $|0\rangle, \ldots, |2^n - 1\rangle$. If each $\alpha_i = \frac{1}{\sqrt{2^n}}$, then $|\psi\rangle$ is said to be in a *uniform superposition*.

To simplify reasoning about qubits in complex systems, we often group them into *registers*. A register corresponds to the state space of the qubits it contains, allowing us to

work at a higher level of abstraction. Instead of focusing on individual qubits, we consider unitary operations and measurements that act on the entire state space of a register. For example, suppose we have a multi-qubit state $|\psi\rangle$ in the tensor product space $\mathcal{A} \otimes \mathcal{B}$. We use the same symbols $\mathcal{A}$ and $\mathcal{B}$ to refer both to the state spaces $\mathcal{A}$ and $\mathcal{B}$ as well as to the registers corresponding to the qubits within these state spaces, respectively. We use the same symbols to refer both to the state spaces $\mathcal{A}$ and $\mathcal{B}$ as $\mathcal{A}$ and $\mathcal{B}$, respectively. To make this correspondence explicit, we sometimes add the register as a subscript. For instance, if $|\psi\rangle \in \mathcal{A}$, we may write $|\psi\rangle_{\mathcal{A}}$, and if $|\psi\rangle \in \mathcal{A} \otimes \mathcal{B}$, we may write $|\psi\rangle_{\mathcal{AB}}$. We extend this convention to linear operators as well. If I is the identity operator on $\mathcal{A}$ and U is a unitary operator acting on $\mathcal{B}$, their combined operation on $\mathcal{A} \otimes \mathcal{B}$ is written as $I_{\mathcal{A}} \otimes U_{\mathcal{B}}$.

In many cases, we may want to apply a linear operator to a larger Hilbert space than the one for which it is explicitly defined. For clarity, we adopt the convention that any operator is implicitly understood to act as the tensor product with the identity operator on any unaffected registers. This ensures that operators are well-defined in the context of larger systems while preserving their intended behaviour.

2.2.2 Mixed states and partial trace

So far, we have discussed *pure states*, which are described by normalised vectors $|\psi\rangle$ in a Hilbert space. However, quantum systems are not always in pure states. A more general framework involves *mixed states*, which arise when a quantum system is in a probabilistic mixture of pure states. A mixed state is represented by a *density matrix*, ρ, which is a positive semidefinite operator of unit trace. A mixed state that is in the pure states $|\psi_1\rangle, \ldots, |\psi_m\rangle$ with probabilities $p_1, \ldots, p_m$, respectively, corresponds to the density matrix:

$$\rho = \sum_{i=1}^{m} p_i |\psi_i\rangle\langle\psi_i|.$$

A mixed state ρ is *pure* if and only if its rank is 1, in which case $\rho = |\psi\rangle\langle\psi|$, the density matrix of the pure state $|\psi\rangle$.

Mixed states arise naturally when considering subsystems of entangled states. Suppose we have a quantum state described by a density matrix ρ_{AB} on the tensor product space $\mathcal{A}\otimes\mathcal{B}$. To describe the state of subsystem $\mathcal{A}$ alone, we take the *partial trace* over subsystem $\mathcal{B}$, denoted Tr_B, to obtain the reduced density matrix:

$$\rho_A = \mathrm{Tr}_B(\rho_{AB}) := \sum_b \left(I_{\mathcal{A}} \otimes \langle b|\right) \rho_{AB} \left(I_{\mathcal{A}} \otimes |b\rangle\right), \tag{2.5}$$

where $\{|b\rangle\}$ is an arbitrary orthonormal basis of $\mathcal{B}$. The reduced density matrix ρ_A reproduces the statistics of all possible measurements on the subsystem $\mathcal{A}$, but contains no information about $\mathcal{B}$. If ρ_{AB} is entangled, tracing out part of the entangled system always results in a mixed state, even when ρ_{AB} itself is pure. For example, if we trace out one of the subsystems of the EPR state $|\mathsf{EPR}\rangle$ from (2.4) (on the system $\mathcal{A} \otimes \mathcal{B}$), we obtain

$$\rho_A = \mathrm{Tr}_B(|\mathsf{EPR}\rangle\langle\mathsf{EPR}|) = \frac{1}{2}\left(|0\rangle\langle 0| + |1\rangle\langle 1|\right),$$

which is known as the *maximally mixed state*.

2.2.3 Quantum gates and unitaries

By the postulates of quantum mechanics, any n-qubit state $|\psi\rangle$ can be transformed into another n-qubit state $|\phi\rangle$ via a unitary operation on $\mathbb{C}^{2^n}$. Any such unitary operation U is reversible, with the inverse given by $U^{-1} = U^\dagger$. For $n = 1, 2$, these operations are

commonly referred to as *quantum gates*. Below, we define some *elementary gates*, many of which are used in this work.

By associating the basis states $|0\rangle$ and $|1\rangle$ with the column vectors $[1\ 0]^T$ and $[0\ 1]^T$, respectively, these gates can be conveniently expressed as 2×2 matrices. We begin with the *Pauli matrices*, which act on single qubits:

$$\mathsf{X} := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad \text{flips the computational basis states } |0\rangle \text{ and } |1\rangle,$$

$$\mathsf{Z} := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad \text{adds a } -1 \text{ phase to } |1\rangle,$$

$$\mathsf{Y} := \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \qquad \text{combines } \mathsf{X} \text{ and } \mathsf{Z} \text{ with a factor of } i, \text{ given by } \mathsf{Y} = i\mathsf{XZ}.$$

Another important single-qubit gate is the *Hadamard gate*:

$$\mathsf{H} := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad \text{creates superpositions of } |0\rangle \text{ and } |1\rangle.$$

The Hadamard gate can be applied to all n qubits of the all-zero state $|0^n\rangle$ to create a uniform superposition over all n-qubit computational basis states:

$$\mathsf{H}^{\otimes n}|0^n\rangle = \frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} |i\rangle.$$

A key 2-qubit gate is the *Controlled-NOT gate*:

$$\mathsf{CNOT} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad \text{flips the target qubit if the control qubit is } |1\rangle. \tag{2.6}$$

The CNOT gate is fundamental for generating (or destroying) entanglement. For instance, when combined with the Hadamard gate, it can be used to construct the EPR-state $|\mathsf{EPR}\rangle$ from (2.4):

$$\mathsf{CNOT}\,(\mathsf{H} \otimes I)\,|00\rangle = \mathsf{CNOT}\frac{1}{\sqrt{2}}\,(|00\rangle + |10\rangle) = \frac{1}{\sqrt{2}}\,(|00\rangle + |11\rangle)\,.$$

In addition to generating entanglement, the CNOT gate can copy or uncompute classical information:

$$\mathsf{CNOT}|b\rangle|0\rangle = |b\rangle|b\rangle, \qquad \text{copies the control qubit into the target qubit register,}$$
$$\mathsf{CNOT}|b\rangle|b\rangle = |b\rangle|0\rangle, \qquad \text{uncomputes the target qubit using the control qubit.}$$

It is important to note that no unitary operation can copy a *general* quantum state. This is known as the *no-cloning theorem* [Par70].

Another useful operation is the *Swap gate*, which exchanges the states of any two (multi) qubit registers:

$$\mathsf{SWAP}|\psi\rangle|\phi\rangle = |\phi\rangle|\psi\rangle.$$

Lastly, we introduce *controlled operations*, which apply a unitary transformation conditioned on the state of a control register. For example, given a sequence of unitary operations $U_0, \ldots, U_{2^n-1}$ acting on the same Hilbert space (one for each computational basis state in $\mathbb{C}^{2^n}$), we define

$$U = \sum_{i=0}^{2^n-1} |i\rangle\langle i| \otimes U_i.$$

This operation applies U_i to the second register if the first (control) register is in state $|i\rangle$. The CNOT gate is an instance of a controlled operation:

$$\mathsf{CNOT} = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes \mathsf{X}.$$

2.2.4 Measurements

Apart from unitary transformations, quantum mechanics allows us to perform *measurements*, which enable the extraction of classical information from quantum states. Measurements are inherently probabilistic and governed by *Born's rule*, which states that the probability of observing an outcome is proportional to the square of the amplitude associated with that outcome. After the measurement, the quantum state *collapses* to the state corresponding to the observed outcome.

The most common type of quantum measurement is in the computational basis. For a quantum state

$$|\psi\rangle = \sum_{i\in\{0,1\}^n} \alpha_i |i\rangle,$$

the probability of observing outcome i is $|\alpha_i|^2$. After the measurement, the state collapses to the corresponding basis state $|i\rangle$. This process allows us to extract n-bit of classical information from the n-qubit state $|\psi\rangle$, but it destroys $|\psi\rangle$ in the process.

More generally, we can perform a *projective measurement*, which is described by a set of mutually orthogonal projectors $\{P_1, P_2, \ldots, P_m\}$, which span $\mathbb{C}^{2^n}$. Specifically, this means that the projectors sum to the identity (on $\mathbb{C}^{2^n}$). Each projector P_i corresponds to a subspace of $\mathbb{C}^{2^n}$, and since the projectors are mutually orthogonal, so are the corresponding subspaces. Any quantum state $|\psi\rangle \in \mathbb{C}^{2^n}$ can then be decomposed into components within these subspaces:

$$|\psi\rangle = \sum_{i=1}^{m} P_i|\psi\rangle.$$

The probability of observing outcome i is given by

$$\|P_i|\psi\rangle\|^2 = \langle\psi|P_i|\psi\rangle,$$

after which the state collapses to

$$\frac{P_i|\psi\rangle}{\|P_i|\psi\rangle\|}.$$

The most general type of measurement is known as a *positive operator-valued measurement (POVM)*, which generalises projective measurements. POVMs are particularly useful when we are only interested in the final probability distribution of outcomes and not the resulting post-measurement state itself. A POVM is described by positive semidefinite operators $E_1, E_2, \ldots, E_m$, which sum up to the identity on $\mathbb{C}^{2^n}$. For a quantum state $|\psi\rangle \in \mathbb{C}^{2^n}$, the probability of obtaining outcome i is given by

$$\mathrm{Tr}[E_i|\psi\rangle\langle\psi|] = \langle\psi|E_i|\psi\rangle.$$

POVMs are more general than projective measurements because the operators E_i are not necessarily orthogonal projectors. They can represent incomplete or noisy measurements and are widely used in quantum information theory.

2.2.5 Quantum algorithms

A *quantum algorithm* is a sequence of unitary operations and measurements applied to a quantum system. Without loss of generality, we assume that the quantum algorithm begins with all qubits initialised to the state $|0\rangle$. If the quantum algorithm involves no intermediate measurements, the entire process can be represented as a single unitary operation. This unitary operation can, in turn, be implemented by a *quantum circuit*, which is a sequence of quantum gates. Since all unitary operations are invertible, the inverse of the quantum circuit can also be constructed and executed when needed.

We now discuss two quantum algorithms that are frequently used as subroutines in this work. The first is the unitary operation known as the *quantum Fourier transform (QFT)*:

Definition 2.2.1 (Quantum Fourier Transform)**.** *Let $N \in \mathbb{N}_{\geq 1}$. The N-dimensional quantum Fourier transform, denoted by QFT_N, is a unitary operation that for each $j \in [N-1]_0$ performs the mapping:*

$$\mathsf{QFT}_N : |j\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{\frac{2\pi\iota}{N}jk}|k\rangle =: |\widehat{j}\rangle,$$

where ι denotes the imaginary unit, to avoid ambiguity with the index variable i.

For $N = 2$, QFT_2 corresponds to the Hadamard gate H. The QFT enables the transition between the computational basis $\{|j\rangle\}$ and the *Fourier basis* $\{|\widehat{j}\rangle\}$, serving as a fundamental building block for numerous quantum algorithms. One notable example is *phase estimation*, which we will explore in detail in Section 3.3.

When discussing quantum algorithms, we are especially interested in their *cost*. The cost of a quantum algorithm can be measured in various ways, but a natural measure is to count the number of elementary operations the algorithm performs. This measure is referred to as the *time cost* or *gate cost* of the algorithm. The *time complexity* or *gate complexity* of a problem is the minimal time cost of any algorithm that solves the problem. For example, the N-dimensional quantum Fourier transform defined in Definition 2.2.1 has a time cost of $O(\log(N)^2)$ [Cop02].

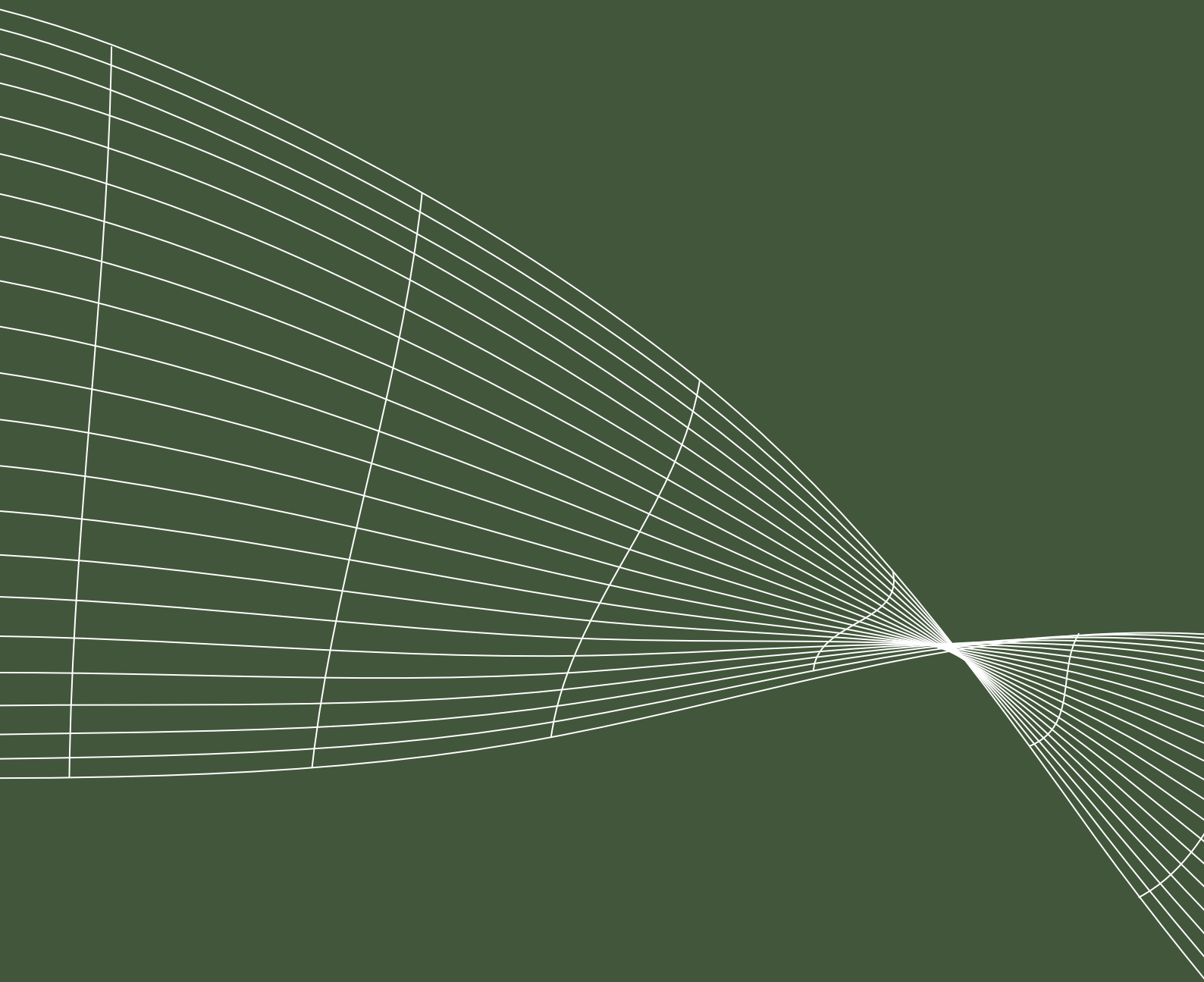
In some cases, algorithms are provided with input through an oracle. For such algorithms, another useful metric is the number of calls, or *queries*, made to the oracle. This is referred to as the *query cost* of the algorithm. By minimising the query cost over all algorithms that solve a given problem, we obtain its *query complexity*. Since the query cost does not take into account the gate cost of any operations that the algorithm performs in between its queries, the query cost is upper bounded by the time cost.

As an example of a problem with an input oracle, consider the *Search* problem. Here the input is an n-bit string $x \in \{0,1\}^n$, and the goal is to output an index $i \in [n]$ such that $x_i = 1$, or to output that no such index exists. The input can be accessed through the oracle $\mathcal{O}_x$, a unitary operator acting on $\mathbb{C}^{[n]\times\{0,1\}}$, that for each $i \in [n]$ and $b \in \{0,1\}$ acts as

$$\mathcal{O}_x|i\rangle|b\rangle \mapsto |i\rangle|b \oplus x_i\rangle.$$

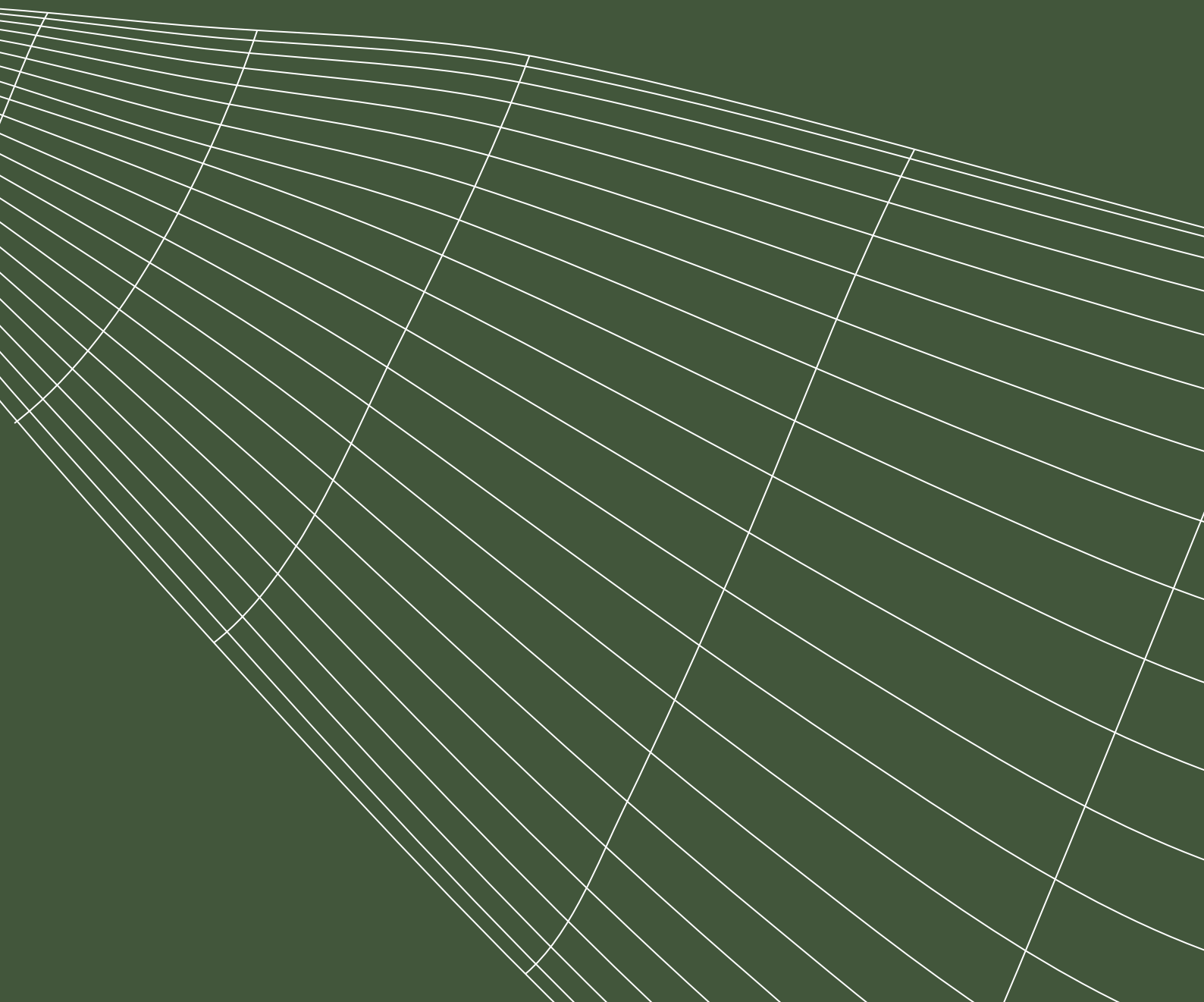
Classically, the query complexity of the Search problem is $\Theta(n)$. Quantumly however, this problem can be solved with bounded error in $O(\sqrt{n})$ queries to $\mathcal{O}_x$ [Gro96], and the time complexity only incurs a minor double logarithmic term [Gro02, ADW17]. This results in a quadratic speedup in the query complexity compared to classical algorithms. Moreover, this speedup is known to be (asymptotically) optimal due to a matching lower bound [BBBV97].

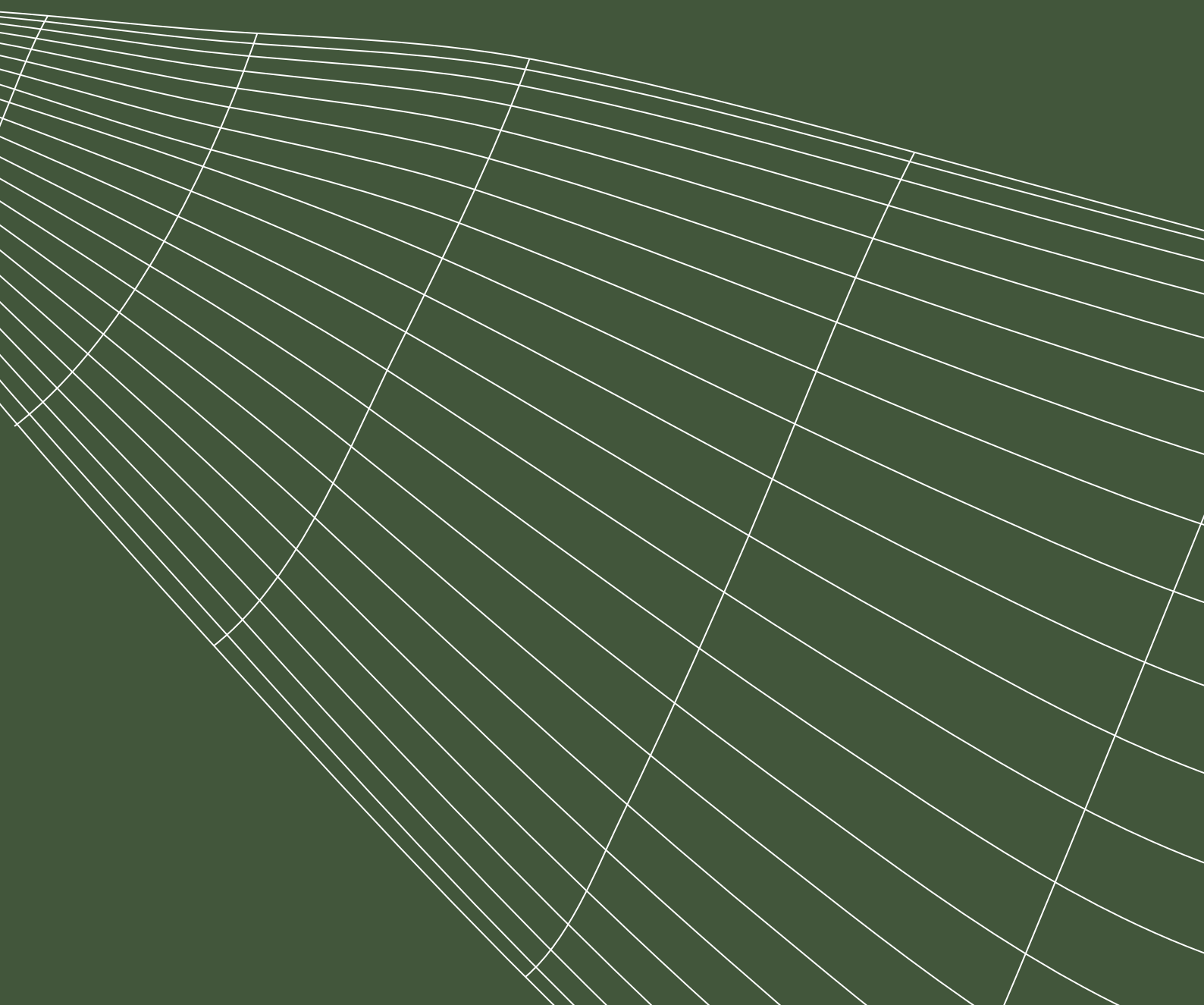
We explore quantum query complexity in more detail in Chapter 7.



Part I

Multidimensional Quantum Walks





Chapter 3

Introduction to quantum walks

Weet je goed hoe hard je wandelt, weet je niet meer waar je bent.

Het Schrödingervergelijkinglied

This chapter is based on Sections 1, 2, and 3 of the paper *Multidimensional Quantum Walks, with Application to k-Distinctness* [JZ23], which is joint work with Stacey Jeffery, and Section 2 of the paper *Multidimensional Electrical Networks and their Application to Exponential Speedups for Graph Problems* [LZ23], which is joint work with Jianqiang Li.

Quantum walk search frameworks are important because they allow the design of quantum algorithms by first constructing a classical random walk algorithm of a particular form, which can then be compiled into a faster quantum algorithm. In this chapter, we introduce such a quantum walk search framework and provide the necessary background to understand the novel concepts and results discussed in Part I of the thesis. We begin by reviewing fundamental graph-theoretic concepts and the basics of electrical networks. Following this, we explore a class of quantum algorithms that utilise phase estimation, known as quantum walks, and examine how the analysis of these algorithms connects back to electrical networks. We then review how these quantum walk algorithms can be applied to locate marked vertices in a graph or to sample from the electrical flow in an electrical network.

3.1 Quantum walk frameworks

Designing quantum algorithms requires a fundamentally different—and often more complex—type of intuition compared to classical algorithms. Developing general design strategies is therefore essential for building this intuition. One such strategy involves the use of quantum walk search frameworks. These frameworks start with a classical random walk and transform it into a quantum algorithm, known as a quantum walk, which often achieves faster run times than its classical counterpart.

The first quantum walk search framework is due to Szegedy [Sze04], and is a generalisation of the technique used by Ambainis in his element distinctness algorithm [Amb07]. The framework can be described in analogy to a classical random walk algorithm that first samples an initial vertex according to the stationary distribution π of some random walk (equivalently, reversible Markov process) P, and repeatedly takes a step of the random walk by sampling a neighbour of the current vertex, checking each time if the current vertex belongs to some *marked set* M. Let $HT(P, M)$ be the hitting time, or the expected number of steps needed by a walker starting from π to reach a vertex in M. If S is the cost of sampling from π, U is the cost of sampling a neighbour of any vertex, C is the cost of checking if a vertex is marked, and $\mathcal{H}$ is an upper bound on $HT(P, M)$ assuming $M \neq \emptyset$, then this classical algorithm finds a marked vertex with bounded error in complexity

$$O(\mathsf{S} + H(\mathsf{U} + \mathsf{C})).$$

Szegedy showed that given such a P and M, if S is the cost of coherently sampling from π, i.e. generating $\sum_u \sqrt{\pi(u)}|u\rangle$, and U is the cost of generating, for any u, the superposition over its neighbours $\sum_v \sqrt{P_{u,v}}|v\rangle$, then there is a quantum algorithm that detects if $M \neq \emptyset$ with bounded error in complexity:

$$O(\mathsf{S} + \sqrt{H}(\mathsf{U} + \mathsf{C})).$$

Although technically the classical S and U might be different from the quantum ones, they are often similar in practice. This result was extended to the case of *finding* a marked vertex, rather than just *detecting* a marked vertex in [AGJK20]. This framework, and subsequent related frameworks have been widely applied, because this is a relatively simple way to design a quantum algorithm.

In [Bel13], Belovs generalised this framework to the *electric network framework*, which we discuss in more detail in Section 3.4. Here the initial state is allowed to be the quantum analog $|\sigma\rangle = \sum_u \sqrt{\sigma(u)}|u\rangle$ of *any* distribution σ, analogous to starting a random walk in some arbitrary initial distribution. Then if S_σ is the cost to generate $|\sigma\rangle$, there is a quantum algorithm that detects a marked vertex with bounded error in complexity:

$$O(\mathsf{S}_\sigma + \sqrt{C}(\mathsf{U} + \mathsf{C})),$$

where C is a quantity that may be the same, or much larger than the hitting time of the classical random walk starting at σ. For example, if $\sigma = \pi$, then $C = H$ as above, but when σ is supported on a single vertex s, and $M = \{t\}$, C is the *commute time* from s to t [CRR+96], which is the expected number of steps needed to get from s to t, and then back to s. If the hitting time from s to t is the same as the hitting time from t to s, this is just twice that hitting time. However, in some cases the hitting time from t to s may be significantly larger than the hitting time from s to t.

The electrical network framework thanks its name due to the fact that the analysis of the quantum walk involves analysing the underlying graph as if it were an electrical network. Here the weighted edges mimic electrical wires, where the weights represent the conductance of the edges. In the case where σ is supported on a single vertex s,

and $M = \{t\}$, we can imagine s being the battery and t the ground. The energy of the electrical current flowing from s to t, i.e. the s-t *electrical flow*, is then related to the quantity C. This connection to electrical networks has later been made even more explicit by [Pid19, AP22], who showed that quantum walks in the electrical network framework not only detect marked vertices but can also generate a quantum state that encodes the propagation of the s-t electrical flow through the graph G.

A second incomparable quantum walk search framework that is similarly easy to apply is the MNRS framework [MNRS11]. Loosely speaking, this is the quantum analogue of a classical random walk that does not check if the current vertex is marked at every step, but rather, only after sufficiently many steps have been taken so that the current vertex is independent of the previously checked vertex. In [AGJ20], the authors extended the electric network framework to be able to *find* a marked vertex, and also showed that the MNRS framework can be seen as a special case of the resulting framework. Thus, the finding version of the electric network framework captures all quantum walk search frameworks in one unified framework.

In this chapter, we focus on quantum walk algorithms as designed in the electric network framework, as well as the later modifications in [Pid19, AP22].

3.2 Preliminaries

Before delving deeper into quantum walks, we first define the necessary graph-theoretic concepts and provide a review of basic knowledge on electrical networks. While experienced readers may already be familiar with these notions, we encourage them not to skip these definitions, as some may differ slightly from the standard terminology used in other works on quantum walks.

3.2.1 Graph theory and electrical networks

Throughout this thesis, our quantum walks are performed on *networks*.

Definition 3.2.1 (Network)**.** *A network is a connected weighted graph $G = (V, E, \mathsf{w})$ with a vertex set V, an (undirected) edge set E and some weight function $\mathsf{w} : E \to \mathbb{R}_{>0}$. Since edges are undirected, we can equivalently describe the edges by some set $\overrightarrow{E}$ such that for all $(u, v) \in E$, exactly one of (u, v) or (v, u) is in $\overrightarrow{E}$. The choice of edge directions is arbitrary. Then we can view the weights as a function $\mathsf{w} : \overrightarrow{E} \to \mathbb{R}_{>0}$, and for all $(u, v) \in \overrightarrow{E}$, define $\mathsf{w}_{v,u} = \mathsf{w}_{u,v}$. For convenience, we define $\mathsf{w}_{u,v} = 0$ for every pair of vertices such that $(u, v) \notin E$. We write*

$$\mathcal{W} := \sum_{(u,v)\in\overrightarrow{E}} \mathsf{w}_{u,v},$$

for the total weight of the network.
For an implicit network G, and $u \in V$, we will let $\Gamma(u)$ denote the neighbourhood *of u:*

$$\Gamma(u) := \{v \in V : (u, v) \in E\}.$$

In case of ambiguity to with respect of which graph G the weighted degree is taken, we write $\Gamma_G(u)$. We use the following notation for the out- and in-neighbourhoods *of $u \in V$:*

$$\begin{aligned} \Gamma^+(u) &:= \{v \in \Gamma(u) : (u, v) \in \overrightarrow{E}\} \\ \Gamma^-(u) &:= \{v \in \Gamma(u) : (v, u) \in \overrightarrow{E}\}, \end{aligned} \tag{3.1}$$

To build intuition from physics and apply results from electrical network theory, it is useful to interpret our networks as *electrical networks*.

Definition 3.2.2 (Electrical network)**.** *Given a network $G = (V, E, \mathsf{w})$ with a weight function w, we can interpret every edge $(u, v) \in E$ as a resistor with resistance $1/\mathsf{w}_{u,v}$. This allows G to be modeled as an* electrical network*.*

Definition 3.2.3 (Flow, Circulation)**.** *A* flow *on a network $G = (V, E, \mathsf{w})$ is a real-valued function $\theta : \overrightarrow{E} \to \mathbb{R}$, extended to edges in both directions by $\theta_{u,v} = -\theta_{v,u}$ for all $(u, v) \in \overrightarrow{E}$. For any flow θ on G and vertex $u \in V$ we define $\theta_u = \sum_{v \in \Gamma(u)} \theta_{u,v}$ as the flow coming out of u. If $\theta_u = 0$, we say flow is conserved at u. If the flow is conserved at every vertex, we call θ a* circulation*. If $\theta_u > 0$, we call u a* source*, and if $\theta_u < 0$ we call u a* sink*. A flow with a unique source s and unique sink t (satisfying $\theta_s = -\theta_t = 1$) is called an* (unit) s-t flow*. The* energy *of any flow θ is*

$$\mathcal{E}(\theta) := \sum_{(u,v) \in \overrightarrow{E}} \frac{\theta_{u,v}^2}{\mathsf{w}_{u,v}}.$$

The effective resistance $\mathcal{R}_{s,t}$ *is given by the minimal energy $\mathcal{E}(\theta)$ over all unit flows θ from s to t. The s-t* electrical flow *is the unique unit s-t flow that achieves this minimal energy.*

Definition 3.2.4 (Potential)**.** *A* potential vector *(also known as potential function) on a network $G = (V, E, \mathsf{w})$ is a real-valued function $\mathsf{p} : V \to \mathbb{R}$ that assigns a potential p_u to each vertex $u \in V$.*

Two fundamental laws governing electrical networks are *Kirchhoff's Law* (also known as Kirchhoff's Node Law) and *Ohm's Law*. Kirchhoff's Law defines an s-t flow as follows:

Definition 3.2.5 (Kirchhoff's Law)**.** *For any given s-t flow θ on an electrical network $G = (V, E, \mathsf{w})$ with $s, t \in V$, the amount of electrical flow entering any vertex $u \in V \backslash \{s, t\}$ must equal the amount of flow exiting u. In other words:*

$$\sum_{v \in \Gamma(u)} \theta_{u,v} = 0.$$

Ohm's Law, on the other hand, states that if a unit of current is injected at s and extracted at t in the electrical network G, then an induced potential vector p is generated, as described in Definition 3.2.4, which is related to the s-t electrical flow θ in the following manner:

Definition 3.2.6 (Ohm's Law)**.** *Let θ be the s-t electrical flow on an electrical network $G = (V, E, \mathsf{w})$ with $s, t \in V$. Then there exists a potential vector p such that the potential difference between the two endpoints of any edge $(u, v) \in E$ is equal to the amount of electrical flow $\theta_{u,v}$ along this edge multiplied with the resistance $1/\mathsf{w}_{u,v}$, that is, $\mathsf{p}_u - \mathsf{p}_v = \theta_{u,v}/\mathsf{w}_{u,v}$.*

The potential p induced by an s-t electrical flow θ in Ohm's Law is not unique. Therefore, it is common practice to consider the potential p that assigns $\mathsf{p}_t = 0$, in which case $\mathsf{p}_s = \mathcal{R}_{s,t}$, where $\mathcal{R}_{s,t}$ is the effective resistance between s and t.

It is often useful to modify our networks by replacing an edge with a longer path. When we modify the network in this way, the flow on the new network is naturally derived from the flow on the original network:

Definition 3.2.7 (Networks with lengths)**.** *If G is a network, and $\ell : \overrightarrow{E} \to \mathbb{Z}_{\geq 1}$ a positive-integer-valued function on the edges of G, we define G^ℓ to be the graph obtained from replacing each edge $(u, v) \in \overrightarrow{E}$ of G with a path from u to v of length $\ell_{u,v}$, and giving each edge in the path the weight $\mathsf{w}_{u,v}$. We define*

$$\mathcal{W}^\ell := \mathcal{W}(G^\ell) = \sum_{e \in \overrightarrow{E}} \mathsf{w}_e \ell_e,$$

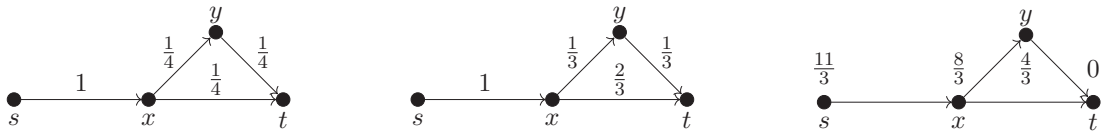


Figure 3.1: Graph G with its s-t electrical flow θ and corresponding potential p at each vertex.

and for any flow θ on G, we let θ^ℓ be the flow on G^ℓ obtained by assigning flow $\theta(u,v)$ to any edge in the path from u to v, and define

$$\mathcal{E}^\ell(\theta) := \mathcal{E}(\theta^\ell) = \sum_{e \in \overrightarrow{E}} \frac{\theta(e)^2}{\mathsf{w}_e} \ell_e.$$

Example graph

To build some intuition for these definitions, we provide an example graph that will be used as a running example throughout Part I of the thesis. Consider the network $G = (V, E, \mathsf{w})$, where the vertex set is given by $V = \{s, x, y, t\}$, and the directed edge set is $\overrightarrow{E} = \{(s,x), (x,y), (x,t), (y,t)\}$. The weight of each edge $(u,v) \in \overrightarrow{E}$ is $\mathsf{w}_{u,v} = \frac{1}{4}$, except for the edge (s,x), which has a weight of $\mathsf{w}_{s,x} = 1$. This network is visualised in Figure 3.1, along with the s-t electrical flow θ on G and the corresponding potential vector p. It is straightforward to verify that the flow θ satisfies Kirchhoff's Law (see Definition 3.2.5), which states that the net flow entering any vertex, except for the source s and the sink t, is zero. Additionally, the potential vector p satisfies Ohm's Law (see Definition 3.2.6), meaning that for each edge (u,v), the potential difference $\mathsf{p}_u - \mathsf{p}_v$ is given by $\frac{\theta_{u,v}}{\mathsf{w}_{u,v}}$. The effective resistance $\mathcal{R}_{s,t}$ can be determined in two ways: either by computing the energy of the flow depicted in Figure 3.1, or by noting that the potential at s is $\mathsf{p}_s = \frac{11}{3}$. Thus, the effective resistance for this network is $\mathcal{R}_{s,t} = \frac{11}{3}$.

3.2.2 The incidence matrix

There is a direct linear relationship between the s-t electrical flow and the potential vector, which can be described by reformulating Kirchhoff's and Ohm's Laws as linear equations involving the incidence matrix of a network G. We follow the treatment in [Vis13, section 4].

Definition 3.2.8 (The edge-vertex incidence matrix)**.** *Let $G = (V, E, \mathsf{w})$ be a network (See Definition 3.2.1). The incidence matrix $B \in \mathbb{C}^{\overrightarrow{E} \times V}$ of G is the matrix whose rows are indexed by $(u,v) \in \overrightarrow{E}$, whose columns are indexed by $u \in V$, and whose non-zero entries are given by*

$$B_{(u,v),u} = \sqrt{\mathsf{w}_{u,v}}, \qquad B_{(u,v),v} = -\sqrt{\mathsf{w}_{u,v}}.$$

Let $W \in \mathbb{C}^{\overrightarrow{E} \times \overrightarrow{E}}$ be the diagonal matrix with entries $W_{(u,v),(u,v)} = \frac{1}{\sqrt{\mathsf{w}_{u,v}}}$. For simplicity, the matrix W can be ignored if we consider the network to be unweighted, i.e. every edge has weight 1. Viewing a flow θ on $G = (V, E, \mathsf{w})$ as a vector in $\mathbb{C}^{\overrightarrow{E}}$, we can multiply it by W to obtain the weighted flow vector $W\theta \in \mathbb{C}^{\overrightarrow{E}}$, where each entry is given by $(W\theta)_{u,v} = \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}$ for each row indexed by $(u,v) \in \overrightarrow{E}$. The norm of $W\theta$ is the square root of the energy of the flow, i.e. $\sqrt{\mathcal{E}(\theta)}$. Introducing $W\theta$ allows us to rephrase Kirchhoff's

Law from Definition 3.2.5 as a linear equation involving the incidence matrix B. Fix an ordering of the columns of B as $s, u_1, \ldots, u_2, t$ for some vertices $u_1, u_2 \in V \setminus \{s, t\}$, and let $\mathsf{e}_i \in \mathbb{C}^n$ be the basis vector with a 1 in the i-th position.

Definition 3.2.9 (Kirchhoff's Law (incidence matrix))**.** *Let θ be any unit s-t flow on an electrical network $G = (V, E, \mathsf{w})$. Let B be the incidence matrix of G. Then θ satisfies*

$$B^T W\theta = \begin{bmatrix} \sum_{v\in\Gamma(s)} \theta_{s,v} \\ \sum_{v\in\Gamma(u_1)} \theta_{u_1,v} \\ \vdots \\ \sum_{v\in\Gamma(u_2)} \theta_{u_2,v} \\ \sum_{v\in\Gamma(t)} \theta_{t,v} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix} = \mathsf{e}_s - \mathsf{e}_t. \tag{3.2}$$

Recall from Definition 3.2.3 that the s-t electrical flow minimises $\mathcal{E}(\theta)$ among all unit s-t flows θ. Since $\mathcal{E}(\theta) = \|W\theta\|^2$, the s-t electrical flow corresponds to the smallest (in norm) solution to (3.2), i.e. the unique s-t flow θ such that $W\theta \in \ker(B^T)^\perp$. We can recover $W\theta$ using the Moore-Penrose inverse of B^T, denoted B^{T+} to manage the double superscript, since the Moore-Penrose inverse A^+ of any linear operator A maps $\mathrm{ran}(A)$ to $\ker(A)^\perp$. Thus, by left-multiplying both sides of (3.2) by B^{T+}, we derive the following property of electrical networks:

Theorem 3.2.10 (Theorem 4.7 in [Vis13])**.** *Let θ be the s-t electrical flow on a network $G = (V, E, \mathsf{w})$. Let B be the incidence matrix of G. Then its flow vector $W\theta$ is given by*

$$W\theta = B^{T+}(\mathsf{e}_s - \mathsf{e}_t). \tag{3.3}$$

Similarly, we can represent a potential vector p as a vector in $\mathbb{C}^V$, with each entry p_u indexed by $u \in V$. This allows us to rephrase Ohm's Law from Definition 3.2.6 as a linear equation involving the incidence matrix B. Fix an ordering of the rows of B as $(u_1, v_1), \ldots, (u_2, v_2) \in \overrightarrow{E}$.

Definition 3.2.11 (Ohm's Law (incidence matrix))**.** *Let θ be the s-t electrical flow on an electrical network $G = (V, E, \mathsf{w})$. Let B be the incidence matrix of G. Then there exists a potential vector p such that*

$$B\mathsf{p} = \begin{bmatrix} \sqrt{\mathsf{w}_{u_1,v_1}}(\mathsf{p}_{u_1} - \mathsf{p}_{v_1}) \\ \vdots \\ \sqrt{\mathsf{w}_{u_2,v_2}}(\mathsf{p}_{u_2} - \mathsf{p}_{v_2}) \end{bmatrix} = \begin{bmatrix} \frac{\theta_{u_1,v_1}}{\sqrt{\mathsf{w}_{u_1,v_1}}} \\ \vdots \\ \frac{\theta_{u_2,v_2}}{\sqrt{\mathsf{w}_{u_2,v_2}}} \end{bmatrix} = W\theta. \tag{3.4}$$

It is standard practice to assume the potential vector p satisfies $\mathsf{p}_s = \mathcal{R}_{s,t}$ and $\mathsf{p}_t = 0$, which is easier to visualise from the incidence matrix perspective:

Lemma 3.2.12. *Let θ be the s-t electrical flow on an electrical network $G = (V, E, \mathsf{w})$ with effective resistance $\mathcal{R}_{s,t}$. Then there exists a potential vector p satisfying Ohm's Law such that $\mathsf{p}_s = \mathcal{R}_{s,t}$ and $\mathsf{p}_t = 0$.*

Proof. From the incidence matrix B, we obtain $B^T B$, known as the *weighted Laplacian* of G. It is well known in spectral graph theory (see, e.g., Lemma 2.2 in [Vis13]) that $B^T B$ has a zero eigenvalue with multiplicity 1, and its corresponding eigenvector is $\sum_{u\in V} \mathsf{e}_u$. Since $\ker(B) = \ker(B^T B)$, setting $\mathsf{p}_t = 0$ results in a valid solution to (3.4), and this makes the solution unique. By left-multiplying both sides of (3.3) with $(W\theta)^T$, we find

$$\mathcal{R}_{s,t} = \|W\theta\|^2 = (W\theta)^T B^{T+}(\mathsf{e}_s - \mathsf{e}_t) = \mathsf{p}^T(\mathsf{e}_s - \mathsf{e}_t) = \mathsf{p}_s - \mathsf{p}_t = \mathsf{p}_s \qquad \square.$$

Using the Moore-Penrose inverse, we can recover the potential satisfying Ohm's Law from Lemma 3.2.12. To achieve this, we remove the last column of B and the last row of p, obtaining $\overline{B}$ and $\overline{\mathsf{p}}$, respectively. As shown in the proof of Lemma 3.2.12, this operation is justified since B is not full rank. This modification effectively forces $\mathsf{p}_t = 0$:

$$\mathsf{p} = \begin{bmatrix} \overline{\mathsf{p}} \\ 0 \end{bmatrix} = \begin{bmatrix} \overline{B}^{+} W\theta \\ 0 \end{bmatrix}. \tag{3.5}$$

Finally, combining (3.3) and (3.4), we derive the following linear relation:

$$\mathsf{p} = B^{+}W\theta = B^{+}B^{T+}(\mathsf{e}_s - \mathsf{e}_t) = (\underbrace{B^T B}_{=:L})^{+}(\mathsf{e}_s - \mathsf{e}_t). \tag{3.6}$$

Thus, the potential vector (and similarly, the flow vector due to (3.4)) is the solution of a *Laplacian linear system*.

Example graph

We revisit our example graph from Figure 3.1, but now from the perspective of the incidence matrix. The incidence matrix B (see Definition 3.2.8) of G and the Moore-Penrose inverse B^{T+} of its transpose are given by

$$B = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & \frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & -\frac{1}{2} \\ 0 & 0 & \frac{1}{2} & -\frac{1}{2} \end{bmatrix}, \qquad B^{T+} = \begin{bmatrix} \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} & -\frac{5}{6} & -\frac{1}{6} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{6} & -\frac{5}{6} \\ 0 & 0 & \frac{2}{3} & -\frac{2}{3} \end{bmatrix}, \tag{3.7}$$

and the weighted diagonal matrix W is

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}. \tag{3.8}$$

We can now recover the electrical flow θ shown in Figure 3.1 by applying Theorem 3.2.10:

$$W\theta = \begin{bmatrix} \frac{\theta_{s,x}}{\sqrt{\mathsf{w}_{s,x}}} \\ \frac{\theta_{x,y}}{\sqrt{\mathsf{w}_{x,y}}} \\ \frac{\theta_{x,t}}{\sqrt{\mathsf{w}_{x,t}}} \\ \frac{\theta_{y,t}}{\sqrt{\mathsf{w}_{y,t}}} \end{bmatrix} = B^{T+}(\mathsf{e}_s - \mathsf{e}_t) = \begin{bmatrix} \frac{3}{4} & -\frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} & -\frac{5}{6} & -\frac{1}{6} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{6} & -\frac{5}{6} \\ 0 & 0 & \frac{2}{3} & -\frac{2}{3} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{2}{3} \\ \frac{4}{3} \\ \frac{2}{3} \end{bmatrix}.$$

This recovers the effective resistance $\mathcal{R}_{s,t} = 1 + \frac{4}{9} + \frac{16}{9} + \frac{4}{9} = \frac{11}{3}$. Using (3.5), where $\overline{B}$ and its Moore-Penrose inverse $\overline{B}^{+}$ are

$$\overline{B} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix}, \qquad \overline{B}^{+} = \begin{bmatrix} 1 & \frac{2}{3} & \frac{4}{3} & \frac{2}{3} \\ 0 & \frac{2}{3} & \frac{4}{3} & \frac{2}{3} \\ 0 & -\frac{2}{3} & \frac{2}{3} & \frac{4}{3} \end{bmatrix}, \tag{3.9}$$

we find that the resulting potential equals

$$\mathsf{p} = \begin{bmatrix} \overline{\mathsf{p}} \\ 0 \end{bmatrix} = \begin{bmatrix} \overline{B}^{+} W \theta \\ 0 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 1 & \frac{2}{3} & \frac{4}{3} & \frac{2}{3} \\ 0 & \frac{2}{3} & \frac{4}{3} & \frac{2}{3} \\ 0 & -\frac{2}{3} & \frac{2}{3} & \frac{4}{3} \end{bmatrix} \begin{bmatrix} 1 \\ \frac{2}{3} \\ \frac{4}{3} \\ \frac{2}{3} \end{bmatrix} \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{11}{3} \\ \frac{8}{3} \\ \frac{4}{3} \\ 0 \end{bmatrix}, \tag{3.10}$$

confirming that $\mathsf{p}_s = \mathcal{R}_{s,t} = \frac{11}{3}$.

3.2.3 Accessing the graph G

In computations involving a (classical) random walk on a graph G, it is usually assumed that for any $u \in V$, it is possible to sample a neighbour $v \in \Gamma(u)$ according to the distribution

$$\Pr[v] = \frac{\mathsf{w}_{u,v}}{\mathsf{w}_u} \text{ where } \mathsf{w}_u := \sum_{v' \in \Gamma(u)} \mathsf{w}_{u,v'} \text{ is the } \textit{weighted degree of } \mathrm{u}.$$

In case of ambiguity to with respect of which graph G the weighted degree is taken, we write w_u^G.

It is standard to assume this sampling procedure is broken into two steps: (1) sampling some $i \in [d_u]$, where $d_u := |\Gamma(u)|$ is the degree of u, and (2) computing the i-th neighbour of u. That is, we assume that for each $u \in V$, there is an efficiently computable function $f_u : [d_u] \to V$ such that $\mathrm{im}(f_u) = \Gamma(u)$, and we call $f_u(i)$ the *i-th neighbour of u*. In the quantum case (see Definition 3.2.13 below), we assume that the sample (1) can be done coherently, and we use a reversible version of the map $(u, i) \mapsto f_u(i)$. We will also find it convenient to suppose the indices i of the neighbours of u come from some more general set $L(u)$, which may equal $[d_u]$, or some other convenient set, which we call the *edge labels of u*. It is possible to have $|L(u)| > |\Gamma(u)| = d_u$, meaning that some elements of $L(u)$ do not label an edge adjacent to u (these labels should be sampled with probability 0). We assume we have a partition of $L(u)$ into disjoint $L^+(u)$ and $L^-(u)$ such that

$$L^+(u) \supseteq \{i \in L(u) : (u, f_u(i)) \in \overrightarrow{E}\} = \{i \in L(u) : f_u(i) \in \Gamma^+(u)\}$$
$$L^-(u) \supseteq \{i \in L(u) : (f_u(i), u) \in \overrightarrow{E}\} = \{i \in L(u) : f_u(i) \in \Gamma^-(u)\}.$$

Note that for any $(u, v) \in \overrightarrow{E}$, with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$, any of (u, v), (v, u), (u, i), or (v, j) fully specify the edge. Thus, it will be convenient to denote the weight of the edge using any of the alternatives:

$$\mathsf{w}_{u,v} = \mathsf{w}_{v,u} = \mathsf{w}_{u,i} = \mathsf{w}_{v,j}.$$

For any $i \in L(u)$, we set $\mathsf{w}_{u,i} = 0$ if and only if $(u, f_u(i)) \notin E$.

Definition 3.2.13 (Quantum Walk access to G)**.** *For each $u \in V$, let $L(u) = L^+(u) \cup L^-(u)$ be some finite set of* edge labels*, and $f_u : L(u) \to V$ a function such that $\Gamma(u) \subseteq \mathrm{im}(f_u)$. A quantum algorithm has* quantum walk access *to G if it has access to the following subroutines:*

- *A subroutine that "samples" from $L(u)$ by implementing a map $U_\star$ in cost $\mathsf{A}_\star$ that acts as*

$$U_\star|u, 0\rangle = \frac{1}{\sqrt{\mathsf{w}_u}} \left(\sum_{i \in L^+(u)} \sqrt{\mathsf{w}_{u,i}} |u, i\rangle - \sum_{i \in L^-(u)} \sqrt{\mathsf{w}_{u,i}} |u, i\rangle \right) =: |\psi_\star(u)\rangle.$$

- *A subroutine that implements the* transition map $|u,i\rangle \mapsto |v,j\rangle$ *(possibly with some error) where* $i = f_u^{-1}(v)$ *and* $j = f_v^{-1}(u)$*, with costs* $\{\mathsf{T}_{u,i} = \mathsf{T}_{u,v}\}_{(u,v)\in\overrightarrow{E}}$.
- *Query access to the total vertex weights* $\mathsf{w}_u = \sum_{v\in\Gamma(u)} \mathsf{w}_{u,v}$.

We call $\{\mathsf{T}_e\}_{e\in\overrightarrow{E}}$ *the set of* transition costs *and* $\mathsf{A}_\star$ *the* cost of generating the star states.

3.3 Phase estimation algorithms

In this section, we formally define a specific type of quantum algorithm that relies on phase estimation [Kit96] and describe the components required to analyse such an algorithm. All algorithms discussed in Part I of the thesis are of this specific form. The quantum algorithm is introduced here in its most general and abstract form. In Section 3.4, we specialise it to the case of quantum walks, providing more concrete intuition for the concepts and objects defined in this section.

Definition 3.3.1 (Parameters of a Phase Estimation Algorithm)**.** *For an implicit input* $x \in \{0,1\}^*$*, fix a finite-dimensional complex inner product space* H*, a unit vector* $|\psi_0\rangle \in H$*, and sets of vectors* $\Psi^{\mathcal{A}}, \Psi^{\mathcal{B}} \subset H$*. We further assume that* $|\psi_0\rangle$ *is orthogonal to every vector in* $\Psi^{\mathcal{B}}$*. Let* $\Pi_{\mathcal{A}}$ *be the orthogonal projector onto* $\mathcal{A} = \mathrm{span}\{\Psi^{\mathcal{A}}\}$*, and similarly for* $\Pi_{\mathcal{B}}$*.*

Throughout Part I of the thesis, Ψ will refer to a *set* of vectors, whereas Hilbert spaces will be denoted by calligraphic capital letters.

The parameters $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$ define a quantum algorithm as follows. Let

$$U_{\mathcal{AB}} = (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I). \tag{3.11}$$

Perform phase estimation of $U_{\mathcal{AB}}$ on initial state $|\psi_0\rangle$ to a certain precision, measure the phase register, and output 1 if the measured phase is 0, and output 0 otherwise. Theorem 3.3.8 at the end of this section describes what is meant precisely by "perform phase estimation", which precision is sufficient, and when *we can* expect the output to be 1 and when 0.

In practice, unitaries like $U_{\mathcal{AB}}$ that are the product of two reflections are nice to work with, because if each of $\Psi^{\mathcal{A}}$ and $\Psi^{\mathcal{B}}$ is a pairwise orthogonal set, implementing $U_{\mathcal{AB}}$ can be reduced to generating the states in $\Psi^{\mathcal{A}}$ and $\Psi^{\mathcal{B}}$, respectively. A product of reflections has sufficient structure to analyse the relevant eigenspaces, as will become clear throughout this section. Although such a reduction is well-known to be true, we show it formally in Claim 4.3.7

3.3.1 Negative analysis

The first of the two cases we want to distinguish with a phase estimation algorithm is the *negative case*, in which there exists a negative witness. The vectors forming the negative witness, and later the vector forming the positive witness, will be denoted by kets. However, it is important to note that these vectors are not necessarily normalised. The negative witness is defined as follows:

Definition 3.3.2 (Negative Witness)**.** *A* δ-negative witness *for* $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$ *is a pair of vectors* $|w_{\mathcal{A}}\rangle, |w_{\mathcal{B}}\rangle \in H$ *such that* $|\psi_0\rangle = |w_{\mathcal{A}}\rangle + |w_{\mathcal{B}}\rangle$*; and* $|w_{\mathcal{A}}\rangle$ *is mostly in the space* $\mathcal{A}$*, and* $|w_{\mathcal{B}}\rangle$ *is mostly in the space* $\mathcal{B}$*, in the sense that* $\|(I - \Pi_{\mathcal{A}})|w_{\mathcal{A}}\rangle\|^2 \leq \delta$ *and* $\|(I - \Pi_{\mathcal{B}})|w_{\mathcal{B}}\rangle\|^2 \leq \delta$*.*

For intuition, it is useful to think of the case when $\delta = 0$. In that case, there exists a 0-negative witness precisely when $|\psi_0\rangle \in \mathcal{A} + \mathcal{B} = (\mathcal{A}^\perp \cap \mathcal{B}^\perp)^\perp$. For the rest of this chapter, we write Λ_Θ for the orthogonal projector onto the span of the $e^{i\theta}$-eigenspaces of $U_{\mathcal{AB}}$ with $|\theta| \leq \Theta$. The negative analysis relies on the effective spectral gap lemma:

Lemma 3.3.3 (Effective Spectral Gap Lemma [LMR+11]). *Fix $\Theta \in (0, \pi)$. If $|\psi_{\mathcal{A}}\rangle \in \mathcal{A}$, then*

$$\|\Lambda_\Theta(I - \Pi_{\mathcal{B}})|\psi_{\mathcal{A}}\rangle\| \leq \frac{\Theta}{2} \||\psi_{\mathcal{A}}\rangle\|.$$

Lemma 3.3.4 (Negative Analysis). *Fix $\delta \geq 0$ and $\Theta \in (0, \pi)$. Suppose there exists a δ-negative witness, $|w_{\mathcal{A}}\rangle$, $|w_{\mathcal{B}}\rangle$, for $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$. Then we have*

$$\|\Lambda_\Theta|\psi_0\rangle\| \leq \frac{\Theta}{2} \||w_{\mathcal{A}}\rangle\| + 2\sqrt{\delta}.$$

Proof. We can apply the effective spectral gap lemma to $\Pi_{\mathcal{A}}|w_{\mathcal{A}}\rangle \in \mathcal{A}$, to get

$$\begin{aligned}
\frac{\Theta}{2} \||w_{\mathcal{A}}\rangle\| &\geq \frac{\Theta}{2} \|\Pi_{\mathcal{A}}|w_{\mathcal{A}}\rangle\| \geq \|\Lambda_\Theta \left(I - \Pi_{\mathcal{B}} - (I - \Pi_{\mathcal{B}})(I - \Pi_{\mathcal{A}})\right)|w_{\mathcal{A}}\rangle\| \\
&\geq \|\Lambda_\Theta(I - \Pi_{\mathcal{B}})|w_{\mathcal{A}}\rangle\| - \|\Lambda_\Theta(I - \Pi_{\mathcal{B}})(I - \Pi_{\mathcal{A}})|w_{\mathcal{A}}\rangle\| \\
&\geq \|\Lambda_\Theta(I - \Pi_{\mathcal{B}})(|\psi_0\rangle - |w_{\mathcal{B}}\rangle)\| - \|(I - \Pi_{\mathcal{A}})|w_{\mathcal{A}}\rangle\| \\
&\geq \|\Lambda_\Theta(I - \Pi_{\mathcal{B}})|\psi_0\rangle\| - \|\Lambda_\Theta(I - \Pi_{\mathcal{B}})|w_{\mathcal{B}}\rangle\| - \|(I - \Pi_{\mathcal{A}})|w_{\mathcal{A}}\rangle\|.
\end{aligned}$$

Since $|\psi_0\rangle$ is orthogonal to $\mathcal{B}$ by construction, and $\|(I - \Pi_{\mathcal{A}})|w_{\mathcal{A}}\rangle\| \leq \sqrt{\delta}$ and similarly for $\mathcal{B}$, the lemma follows. □

3.3.2 Positive analysis:

We want to distinguish the case where there exists a negative witness (the negative case) from the *positive case*, which is the case where there exists a positive witness, defined as follows:

Definition 3.3.5 (Positive Witness). *A δ-positive witness for $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$ is a vector $|w\rangle \in H$ such that $\langle\psi_0|w\rangle \neq 0$ and $|w\rangle$ is almost orthogonal to all $|\psi\rangle \in \Psi^{\mathcal{A}} \cup \Psi^{\mathcal{B}}$, in the sense that $\|\Pi_{\mathcal{A}}|w\rangle\|^2 \leq \delta \||w\rangle\|^2$ and $\|\Pi_{\mathcal{B}}|w\rangle\|^2 \leq \delta \||w\rangle\|^2$.*

Again, for intuition, we consider the case where $\delta = 0$. A 0-positive witness is exactly a component of $|\psi_0\rangle$ in $(\mathcal{A}+\mathcal{B})^\perp$, which exists precisely when $|\psi_0\rangle \notin \mathcal{A}+\mathcal{B}$. Thus, the case where there exists a 0-positive witness is the complement of the case where there exists a 0-negative witness, so it is theoretically possible to distinguish these two cases. When $\delta > 0$, the two cases may or may not be distinct, depending on δ, and the overlap between $\mathcal{A}$ and $\mathcal{B}$.

When $|w\rangle$ is a 0-positive witness, it is straightforward to see that

$$\|\Lambda_0|\psi_0\rangle\| \geq \frac{|\langle w|\psi_0\rangle|}{\||w\rangle\|},$$

where Λ_0 is the orthogonal projector onto the $(+1)$-eigenspace of $U_{\mathcal{AB}}$. For the case of $\delta > 0$, we need the following lemma, analogous to the effective spectral gap lemma.

Lemma 3.3.6 (Effectively Zero Lemma). *Fix $\delta \geq 0$ and $\Theta \in (0, \pi)$. For $|\psi\rangle \in H$ such that $\|\Pi_{\mathcal{A}}|\psi\rangle\|^2 \leq \delta \||\psi\rangle\|^2$ and $\|\Pi_{\mathcal{B}}|\psi\rangle\|^2 \leq \delta \||\psi\rangle\|^2$,*

$$\|(I - \Lambda_\Theta)|\psi\rangle\|^2 \leq \frac{4\pi^2\delta \||\psi\rangle\|^2}{\Theta^2}.$$

Proof. Let $\{\theta_j\}_{j\in J} \subset (-\pi, \pi]$ be the set of eigenphases of $U_{\mathcal{AB}}$, and let Π_j be the orthogonal projector onto the $e^{i\theta_j}$-eigenspace of $U_{\mathcal{AB}}$, so we can write

$$U_{\mathcal{AB}} = \sum_{j\in J} e^{i\theta_j}\Pi_j. \tag{3.12}$$

We have (see (3.11))

$$U_{\mathcal{AB}}|\psi\rangle = |\psi\rangle + 4\Pi_{\mathcal{A}}\Pi_{\mathcal{B}}|\psi\rangle - 2\Pi_{\mathcal{A}}|\psi\rangle - 2\Pi_{\mathcal{B}}|\psi\rangle, \tag{3.13}$$

and using the triangle inequality, $\|\Pi_{\mathcal{A}}|\psi\rangle\|^2 \leq \delta \||\psi\rangle\|^2$ and $\|\Pi_{\mathcal{B}}|\psi\rangle\|^2 \leq \delta \||\psi\rangle\|^2$, we can compute

$$\begin{aligned} &\|4\Pi_{\mathcal{A}}\Pi_{\mathcal{B}}|\psi\rangle - 2\Pi_{\mathcal{A}}|\psi\rangle - 2\Pi_{\mathcal{B}}|\psi\rangle\|^2 = \|2(2\Pi_{\mathcal{A}} - I)\Pi_{\mathcal{B}}|\psi\rangle - 2\Pi_{\mathcal{A}}|\psi\rangle\|^2 \\ &\leq (\|2(2\Pi_{\mathcal{A}} - I)\Pi_{\mathcal{B}}|\psi\rangle\| + \|2\Pi_{\mathcal{A}}|\psi\rangle\|)^2 \leq 16\delta \||\psi\rangle\|^2 . \end{aligned} \tag{3.14}$$

Thus, by (3.13) and (3.14) and the fact that $|e^{i\theta_j} - 1|^2 = 4\sin^2\frac{\theta_j}{2}$ we can conclude that

$$\begin{aligned} 16\delta \||\psi\rangle\|^2 &\geq \|U_{\mathcal{AB}}|\psi\rangle - |\psi\rangle\|^2 \geq \sum_{j\in J} |e^{i\theta_j} - 1|^2 \|\Pi_j|\psi\rangle\|^2 \\ &\geq 4\sin^2\frac{\Theta}{2} \sum_{j\in J:|\theta_j|>\Theta} \|\Pi_j|\psi\rangle\|^2 \geq 4\sin^2\frac{\Theta}{2} \|(I - \Lambda_\Theta)|\psi\rangle\|^2 . \end{aligned}$$

Then since $\sin^2\frac{\Theta}{2} \geq \frac{4}{\pi^2}\frac{\Theta^2}{4}$ whenever $\Theta \in (-\pi, \pi)$, the result follows. □

Lemma 3.3.7 (Positive Analysis)**.** *Fix $\delta \geq 0$ and $\Theta \in (0, \pi)$. Suppose there exists a δ-positive witness $|w\rangle$ for $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$. Then, letting Λ_Θ be as in Lemma 3.3.3,*

$$\|\Lambda_\Theta|\psi_0\rangle\| \geq \frac{|\langle\psi_0|w\rangle|}{\||w\rangle\|} - \frac{2\sqrt{\delta}\pi}{\Theta}.$$

Proof. We compute

$$\begin{aligned} |\langle\psi_0|\Lambda_\Theta|w\rangle| &\geq |\langle\psi_0|w\rangle| - |\langle\psi_0|(I - \Lambda_\Theta)|w\rangle| && \text{by the triangle ineq.} \\ &\geq |\langle\psi_0|w\rangle| - \||\psi_0\rangle\| \cdot \|(I - \Lambda_\Theta)|w\rangle\| && \text{by Cauchy-Schwarz} \\ &\geq |\langle\psi_0|w\rangle| - \frac{2\sqrt{\delta}\pi}{\Theta} \||w\rangle\| , \end{aligned}$$

where we used $\||\psi_0\rangle\| = 1$ and Lemma 3.3.6. Then:

$$\|\Lambda_\Theta|\psi_0\rangle\| \geq \frac{|\langle w|\Lambda_\Theta|\psi_0\rangle|}{\|\Lambda_\Theta|w\rangle\|} \geq \frac{|\langle\psi_0|w\rangle| - \frac{2\sqrt{\delta}\pi}{\Theta}\||w\rangle\|}{\||w\rangle\|} = \frac{|\langle\psi_0|w\rangle|}{\||w\rangle\|} - \frac{2\sqrt{\delta}\pi}{\Theta}.$$ □

3.3.3 The algorithm

By Lemma 3.3.7, if there exists a δ-positive witness, which happens precisely when there is some component of $|\psi_0\rangle$ that is nearly orthogonal to $\mathcal{A} + \mathcal{B}$, then $|\psi_0\rangle$ overlaps the $e^{i\theta}$-eigenspaces of $U_{\mathcal{AB}}$ for small θ, say with $|\theta| \leq \Theta_0$ for some small-ish choice of Θ_0. The precise overlap depends on the size of this component, and allows us to lower bound the probability that phase estimation of $U_{\mathcal{AB}}$ on $|\psi_0\rangle$ will result in a 0 in the phase register. On the other hand, if $|\psi_0\rangle$ is actually in $\mathcal{A}+\mathcal{B}$, then Lemma 3.3.4 upper bounds the overlap of $|\psi_0\rangle$ with small phase spaces, where "small" is determined by the parameter $\Theta > \Theta_0$. This allows us to upper bound the probability that phase estimation of $U_{\mathcal{AB}}$ on $|\psi_0\rangle$, to precision Θ, will result in a 0 in the phase register. The key is then to choose the parameter Θ small enough so that there is a constant gap between the lower bound on the probability of a 0 phase in the positive case, and the upper bound on the probability of a 0 phase in the negative case. This leads to the following theorem:

Theorem 3.3.8. *For an implicit input $x \in \{0,1\}^*$, fix the parameters of a phase estimation algorithm $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$ as in Definition 3.3.1. Suppose we can generate the state $|\psi_0\rangle$ in cost* S*, and implement $U_{\mathcal{AB}} = (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I)$ in cost* A*.*
Let $c_+ \in [1, 50]$ be some constant, and let $\mathcal{C}_- \geq 1$ be a positive real number that may scale with $|x|$. Let δ and δ' be positive real parameters such that

$$\delta \leq \frac{1}{(8c_+)^3 \pi^8 \mathcal{C}_-} \quad \text{and} \quad \delta' \leq \frac{3}{4} \frac{1}{\pi^4 c_+}.$$

Suppose we are guaranteed that exactly one of the following holds:

Positive Condition: *There is a δ-positive witness $|w\rangle$ (see Definition 3.3.5), s.t.*

$$\frac{|\langle w|\psi_0\rangle|^2}{\||w\rangle\|^2} \geq \frac{1}{c_+}.$$

Negative Condition: *There is a δ'-negative witness $|w_{\mathcal{A}}\rangle, |w_{\mathcal{B}}\rangle$ (Definition 3.3.2), s.t.*

$$\||w_{\mathcal{A}}\rangle\|^2 \leq \mathcal{C}_-.$$

Suppose we perform $T = \sqrt{8}\pi^4 c_+ \sqrt{\mathcal{C}_-}$ steps of phase estimation of $U_{\mathcal{AB}}$ on initial state $|\psi_0\rangle$, and output 1 if and only if the measured phase is 0, otherwise we output 0. Then

Positive Case: *If the positive condition holds, the algorithm outputs 1 with probability at least $\frac{2.25}{\pi^2 c_+} \geq \frac{2.25}{50\pi^2}$.*

Negative Case: *If the negative condition holds, the algorithm outputs 1 with probability at most $\frac{2}{\pi^2 c_+}$.*

Thus, the algorithm distinguishes between these two cases with bounded error, in cost

$$O\left(\mathsf{S} + \sqrt{\mathcal{C}_-}\mathsf{A}\right).$$

Proof. Let $\{\theta_j\}_{j \in J} \subset (-\pi, \pi]$ be the set of eigenphases of $U_{\mathcal{AB}}$, and let Π_j be the orthogonal projector onto the $e^{i\theta_j}$-eigenspace of $U_{\mathcal{AB}}$, so we can write

$$U_{\mathcal{AB}} = \sum_{j \in J} e^{i\theta_j} \Pi_j.$$

After making a superposition over t from 0 to $T-1$ in the phase register, and applying $U^t_{\mathcal{AB}}$ to the input register conditioned on the phase register, as one does in phase estimation [Kit96], we obtain the following state:

$$\sum_{t=0}^{T-1} \frac{1}{\sqrt{T}} |t\rangle U^t_{\mathcal{AB}} |\psi_0\rangle = \sum_{j \in J} \sum_{t=0}^{T-1} \frac{1}{\sqrt{T}} |t\rangle e^{it\theta_j} \Pi_j |\psi_0\rangle.$$

The phase estimation algorithm then proceeds by applying an inverse Fourier transform, $F_T^\dagger$, to the first register, and then measuring the result, to obtain some $t \in [T-1]_0$. We choose the output bit based on whether $t = 0$ or not. The probability of measuring 0

is

$$
\begin{aligned}
p_0 &:= \left\| \langle 0|F_T^\dagger \otimes I \left(\sum_{j\in J} \sum_{t=0}^{T-1} \frac{1}{\sqrt{T}} |t\rangle e^{it\theta_j} \Pi_j |\psi_0\rangle \right) \right\|^2 \\
&= \left\| \sum_{t=0}^{T-1} \frac{1}{\sqrt{T}} \langle t| \otimes I \left(\sum_{j\in J} \sum_{t=0}^{T-1} \frac{1}{\sqrt{T}} |t\rangle e^{it\theta_j} \Pi_j |\psi_0\rangle \right) \right\|^2 \\
&= \frac{1}{T^2} \left\| \sum_{j\in J} \sum_{t=0}^{T-1} e^{it\theta_j} \Pi_j |\psi_0\rangle \right\|^2 = \frac{1}{T^2} \sum_{j\in J:\theta_j\neq 0} \left| \frac{1-e^{i\theta_j T}}{1-e^{i\theta_j}} \right|^2 \|\Pi_j|\psi_0\rangle\|^2 + \|\Lambda_0|\psi_0\rangle\|^2 \\
&= \frac{1}{T^2} \sum_{j\in J:\theta_j\neq 0} \frac{\sin^2(T\theta_j/2)}{\sin^2(\theta_j/2)} \|\Pi_j|\psi_0\rangle\|^2 + \|\Lambda_0|\psi_0\rangle\|^2 ,
\end{aligned}
\tag{3.15}
$$

since $\left|\sum_{t=0}^{T-1} e^{it\theta}\right| = \left|\frac{1-e^{i\theta T}}{1-e^{i\theta}}\right|$, and $|1-e^{i\theta}|^2 = 4\sin^2(\frac{\theta}{2})$ for any $\theta \in \mathbb{R}$. We will analyse the positive and negative cases one-by-one.

Positive Case: Assume the positive condition holds. The existence of a δ-positive witness allows us to apply Lemma 3.3.7. In the following, we will use the identities $\sin^2(\theta) \leq \theta^2$ for all θ, and $\sin^2(\theta) \geq \frac{4\theta^2}{\pi^2}$ whenever $|\theta| \leq \pi/2$. Let $\Theta_0 = \pi/T$. Continuing from (3.15), we can lower bound the probability of measuring a 0 in the phase register as follows:

$$
\begin{aligned}
p_0 &\geq \frac{1}{T^2} \sum_{j\in J:0<|\theta_j|\leq\Theta_0} \frac{\sin^2(T\theta_j/2)}{\sin^2(\theta_j/2)} \|\Pi_j|\psi_0\rangle\|^2 + \|\Lambda_0|\psi_0\rangle\|^2 \\
&\geq \frac{1}{T^2} \sum_{j\in J:0<|\theta_j|\leq\Theta_0} \frac{4(T\theta_j/2)^2/\pi^2}{(\theta_j/2)^2} \|\Pi_j|\psi_0\rangle\|^2 + \|\Lambda_0|\psi_0\rangle\|^2 \\
&\geq \frac{4}{\pi^2} \|\Lambda_{\Theta_0}|\psi_0\rangle\|^2 \geq \frac{4}{\pi^2} \left(\frac{|\langle\psi_0|w\rangle|}{\||w\rangle\|} - \frac{2\sqrt{\delta}\pi}{\Theta_0} \right)^2 \\
&\geq \frac{4}{\pi^2} \left(\frac{1}{\sqrt{c_+}} - \frac{2\pi T}{\pi} \frac{1}{(8c_+)^{3/2}\pi^4\sqrt{\mathcal{C}_-}} \right)^2 \\
&= \frac{4}{\pi^2} \left(\frac{1}{\sqrt{c_+}} - \frac{2\sqrt{8}\pi^4 c_+\sqrt{\mathcal{C}_-}}{(8c_+)^{3/2}\pi^4\sqrt{\mathcal{C}_-}} \right)^2 = \frac{4}{\pi^2} \left(\frac{3}{4}\right)^2 \frac{1}{c_+} = \frac{2.25}{\pi^2}\frac{1}{c_+} \geq \frac{2.25}{50\pi^2}.
\end{aligned}
\tag{3.16}
$$

Negative Case: Assume the negative condition, which allows us to apply Lemma 3.3.4. In the following, we will use the identities $\sin^2(\theta) \leq \min\{1, \theta^2\}$ for all θ, and $\sin^2(\theta/2) \geq \frac{\theta^2}{\pi^2}$ whenever $|\theta| \leq \pi$. Let $\Theta = \pi^{-2}(c_+\mathcal{C}_-)^{-1/2}$. Continuing from (3.15), we can upper bound the probability of measuring a 0 in the phase register:

$$
\begin{aligned}
p_0 &\leq \frac{1}{T^2} \sum_{j\in J:0<|\theta_j|\leq\Theta} \frac{(T\theta_j/2)^2}{(\theta_j/\pi)^2} \|\Pi_j|\psi_0\rangle\|^2 + \frac{1}{T^2} \sum_{j\in J:|\theta_j|>\Theta} \frac{1}{(\theta_j/\pi)^2} \|\Pi_j|\psi_0\rangle\|^2 + \|\Lambda_0|\psi_0\rangle\|^2 \\
&\leq \frac{\pi^2}{4} \|\Lambda_\Theta|\psi_0\rangle\|^2 + \frac{1}{T^2}\frac{\pi^2}{\Theta^2} \leq \frac{\pi^2}{4} \left(\frac{\Theta}{2} \||w_\mathcal{A}\rangle\| + 2\sqrt{\delta'} \right)^2 + \frac{1}{8\pi^8 c_+^2 \mathcal{C}_-}\frac{\pi^2}{\Theta^2} \\
&\leq \frac{\pi^2}{4} \left(\frac{1}{\pi^2\sqrt{c_+\mathcal{C}_-}}\sqrt{\mathcal{C}_-} + 2\frac{\sqrt{3}}{2\pi^2\sqrt{c_+}} \right)^2 + \frac{\pi^2}{8\pi^8 c_+^2\mathcal{C}_-}\pi^4 c_+\mathcal{C}_- \\
&\leq \frac{1}{4\pi^2 c_+} \left(1+\sqrt{3}\right)^2 + \frac{1}{8\pi^2 c_+} \leq \frac{2}{\pi^2 c_+}.
\end{aligned}
$$

To complete the proof, it is easily verified that the described algorithm has the claimed cost. □

3.3.4 Approximating the optimal positive witness

Consider the case where we have a 0-positive witness, meaning $\delta = 0$ in Definition 3.3.5. There are infinitely many 0-positive witnesses, but there exists an unique *optimal positive witness*:

Definition 3.3.9 (Optimal Positive Witness)**.** *Let $|w\rangle \in H$ be a 0-positive witness for $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$ s.t. $\langle w|\psi_0\rangle = 1$. We say that $|w\rangle$ is the optimal positive witness for $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$ if*

$$\||w\rangle\| = \min\{\||w\rangle\| : |w\rangle \text{ is a 0-positive witness for } (H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}}); \langle w|\psi_0\rangle = 1\}.$$

Fact 3.3.10. *The optimal positive witness for $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$ is the unique 0-positive witness that lies in*

$$(I - \Pi_{\mathcal{B}})\Psi_{\mathcal{A}} + \mathrm{span}\{|\psi_0\rangle\},$$

meaning it can be decomposed as

$$|\psi_0\rangle = \frac{\langle w|\psi_0\rangle}{\||w\rangle\|}\frac{|w\rangle}{\||w\rangle\|} + (I - \Pi_{\mathcal{B}})|w_{\mathcal{A}}\rangle$$

for some (unnormalised) $|w_{\mathcal{A}}\rangle \in \Psi^{\mathcal{A}}$.

Proof. Let $|w\rangle$ be the optimal positive witness for $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$ and suppose the contrary holds, meaning $|w\rangle$ contains a component $|w_1\rangle$ that is orthogonal to both $(I-\Pi_{\mathcal{B}})\Psi_{\mathcal{A}}$ as well as $|\psi_0\rangle$. Now consider the state $|w_0\rangle := |w\rangle - |w_1\rangle$, which then must lie in $(I-\Pi_{\mathcal{B}})\Psi_{\mathcal{A}}+\mathrm{span}\{|\psi_0\rangle\}$. Firstly, since by definition $\Pi_{\mathcal{B}}|\psi_0\rangle = 0$ (see Definition 3.3.1) and $\Pi_{\mathcal{B}}|w\rangle = 0$ (see Definition 3.3.5), we find that $\Pi_{\mathcal{B}}|w_0\rangle = 0$ and hence also $\Pi_{\mathcal{B}}|w_1\rangle = 0$. Secondly, we have $\langle\psi_0|w_0\rangle = \langle\psi_0|w\rangle = 1$, since $|w_1\rangle$ is orthogonal to $|\psi_0\rangle$. Lastly, since $|w_1\rangle$ is orthogonal to $(I - \Pi_{\mathcal{B}})\Psi_{\mathcal{A}}$ and by definition $\Pi_{\mathcal{A}}|w\rangle = 0$ (see Definition 3.3.5), we also have

$$\Pi_{\mathcal{A}}|w_0\rangle = \Pi_{\mathcal{A}}|w\rangle - \Pi_{\mathcal{A}}|w_1\rangle = 0 - \Pi_{\mathcal{A}}(I - \Pi_{\mathcal{B}})|w_1\rangle = 0.$$

This means that the component $|w_0\rangle$ of $|w\rangle$ satisfies all the requirements to be a 0-positive witness for $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$ itself and its size is strictly smaller than the size of $|w\rangle$, which is in contradiction with $|w\rangle$ being the optimal positive witness.

The uniqueness of the optimal positive witness now follows as well: suppose that there are two optimal positive witnesses $|w_1\rangle$ and $|w_2\rangle$ for $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$. We have just shown that both $|w_1\rangle$ and $|w_2\rangle$ – and hence also their sum – lie in $(I - \Pi_{\mathcal{B}})\Psi_{\mathcal{A}} + \mathrm{span}\{|\psi_0\rangle\}$. This means specifically that

$$(\langle w_1| + \langle w_2|)(|w_1\rangle - |w_2\rangle) = \||w_1\rangle\|^2 + \||w_2\rangle\|^2 = 0.$$

So either $|w_1\rangle$ or $|w_2\rangle$ has a component in the complement of $(I - \Pi_{\mathcal{B}})\Psi_{\mathcal{A}} + \mathrm{span}\{|\psi_0\rangle\}$, or $|w_1\rangle = |w_2\rangle$. Since the former leads to a contradiction, the latter must be true. □

If we are promised to have a 0-positive witness, meaning $\delta = 0$, then the phase estimation algorithm of Theorem 3.3.8 provides us with more information then simply a 1-bit confirmation of that fact that the positive condition holds, i.e. that a 0-positive witness exists. In fact, the other registers of the phase estimation algorithm contain an approximation of the optimal positive witness and we can obtain a better approximation by running more steps of phase estimation.

Lemma 3.3.11 (Modified Lemma 8 in [Pid19] and Lemma 10 in [AP22])**.** *Fix a phase estimation algorithm $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$ as in Definition 3.3.1. Suppose we can generate the state $|\psi_0\rangle$ in cost* S*, and implement $U_{\mathcal{AB}} = (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I)$ in cost* A*. Let p be a positive real number that may scale with $|x|$ and suppose that $|w\rangle$ is the optimal positive witness (see Definition 3.3.9 and Fact 3.3.10) such that*

$$|\psi_0\rangle = \sqrt{p}\frac{|w\rangle}{\||w\rangle\|} + (I - \Pi_{\mathcal{B}})|w_{\mathcal{A}}\rangle.$$

Then performing T steps of phase estimation of $U_{\mathcal{AB}}$ on initial state $|\psi_0\rangle$ will output that the measured phase is "0" with probability $p_0 \in [p, p + \frac{17\pi^2\||w_{\mathcal{A}}\rangle\|}{16T}]$, leaving a state $|\widetilde{w}\rangle$ satisfying

$$\frac{1}{2}\left\||\widetilde{w}\rangle\langle\widetilde{w}| - \frac{|w\rangle\langle w|}{\||w\rangle\|^2}\right\|_1 \leq \sqrt{1 - \frac{p}{p_0}} \leq \sqrt{\frac{17\pi^2\,\||w_{\mathcal{A}}\rangle\|}{16Tp}}.$$

Proof. Due to (3.16), we know that we lower bound the probability of success by $\|\Lambda_0|\psi_0\rangle\|$. Since we are dealing with a 0-positive witness, we hence know that

$$p_0 \geq \|\Lambda_0|\psi_0\rangle\|^2 \geq \frac{|\langle\psi_0|w\rangle|}{\||w\rangle\|} = \sqrt{p}.$$

We note that the second inequality in the above equation is in fact an equality, due to Lemma 3.3.3:

$$\|\Lambda_0|\psi_0\rangle\|^2 \leq \left\|\Lambda_0\left(|\psi_0\rangle - \sqrt{p}\frac{|w\rangle}{\||w\rangle\|}\right)\right\|^2 + \left\|\Lambda_0\sqrt{p}\frac{|w\rangle}{\||w\rangle\|}\right\|^2 = \left\|\Lambda_0\sqrt{p}\frac{|w\rangle}{\||w\rangle\|}\right\|^2 \leq p. \quad (3.17)$$

For the upper bound, we continue where we left off in (3.15) from the proof of Theorem 3.3.8:

$$p_0 = \frac{1}{T^2}\sum_{j\in J:\theta_j\neq 0}\frac{\sin^2(T\theta_j/2)}{\sin^2(\theta_j/2)}\|\Pi_j|\psi_0\rangle\|^2 + \|\Lambda_0|\psi_0\rangle\|^2. \quad (3.18)$$

We split this sum into two parts, depending on whether the value of θ_j is smaller or larger than $\sqrt{1/(T\,\||w_{\mathcal{A}}\rangle\|)}$. In both parts, we make use of the identity $\sin^2(\theta) \leq \min\{1, \theta^2\}$ for all θ and we abbreviate $|\varphi\rangle = \frac{|w\rangle}{\||w\rangle\|}$. By inserting the same resolution of the identity as in (3.17), we upper bound the first sum as

$$\begin{aligned}
&\frac{1}{T^2}\sum_{j\in J:0<\theta_j<\sqrt{1/(T\||w_{\mathcal{A}}\rangle\|)}}\frac{\sin^2(T\theta_j/2)}{\sin^2(\theta_j/2)}\|\Pi_j|\psi_0\rangle\|^2 && (3.19)\\
&\leq \frac{1}{T^2}\sum_{j\in J:0<\theta_j<\sqrt{1/(T\||w_{\mathcal{A}}\rangle\|)}}\frac{\sin^2(T\theta_j/2)}{\sin^2(\theta_j/2)}\left(\|\Pi_j\left(|\psi_0\rangle - \sqrt{p}|\varphi\rangle\right)\|^2 + \|\Pi_j\sqrt{p}|\varphi\rangle\|^2\right)\\
&= \frac{1}{T^2}\sum_{j\in J:0<|\theta_j|<\sqrt{1/(T\||w_{\mathcal{A}}\rangle\|)}}\frac{\sin^2(T\theta_j/2)}{\sin^2(\theta_j/2)}\|\Pi_j\left(|\psi_0\rangle - \sqrt{p}|\varphi\rangle\right)\|^2\\
&\leq \frac{1}{T^2}\sum_{j\in J:0<|\theta_j|<\sqrt{1/(T\||w_{\mathcal{A}}\rangle\|)}}\frac{\pi^2T^2}{4}\frac{\||w_{\mathcal{A}}\rangle\|}{4T} && \text{by Lemma 3.3.3}\\
&\leq \frac{\pi^2\,\||w_{\mathcal{A}}\rangle\|}{16T}. && (3.20)
\end{aligned}$$

For the second sum, we additionally use the bound $\sin^2(\theta/2) \geq \frac{\theta^2}{\pi^2}$ whenever $|\theta| \leq \pi$:

$$\frac{1}{T^2} \sum_{j\in J:|\theta_j|\geq\sqrt{1/(T\||w_{\mathcal{A}}\rangle\|)}} \frac{\sin^2(T\theta_j/2)}{\sin^2(\theta_j/2)} \|\Pi_j|\psi_0\rangle\|^2 \leq \frac{1}{T^2} \sum_{j\in J:|\theta_j|\geq\sqrt{1/(T\||w_{\mathcal{A}}\rangle\|)}} \frac{\pi^2}{\theta_j^2} \|\Pi_j|\psi_0\rangle\|^2$$
$$\leq \frac{\pi^2 \||w_{\mathcal{A}}\rangle\|}{T} \sum_{j\in J:|\theta_j|\geq\sqrt{1/(T\||w_{\mathcal{A}}\rangle\|)}} \|\Pi_j|\psi_0\rangle\|^2 \leq \frac{\pi^2 \||w_{\mathcal{A}}\rangle\|}{T}. \tag{3.21}$$

By substituting the upper bounds from (3.17),(3.19) and (3.21) into (3.18), we conclude that

$$p_0 \leq \frac{\pi^2 \||w_{\mathcal{A}}\rangle\|}{16T} + \frac{\pi^2 \||w_{\mathcal{A}}\rangle\|}{T} + p = \frac{17\pi^2 \||w_{\mathcal{A}}\rangle\|}{16T} + p.$$

Finally, let $|\widetilde{w}\rangle$ be the (normalised) post measurement state after measuring 0. We abbreviate PE for the phase estimation algorithm followed by the projection onto measuring 0, as described in (3.15), such that $|\widetilde{w}\rangle = \frac{1}{\sqrt{p_0}}\mathsf{PE}|\psi_0\rangle$. Note that since $|\varphi\rangle$ is an 1-eigenvector of U, we have $|\varphi\rangle = \mathsf{PE}|\varphi\rangle$, meaning we can conclude the lemma via the inequality

$$\frac{1}{2} \||\widetilde{w}\rangle\langle\widetilde{w}| - |\varphi\rangle\langle\varphi|\|_1 \leq \sqrt{1 - |\langle\widetilde{w}|\varphi\rangle|^2} = \sqrt{1 - \frac{|\langle\psi_0|\mathsf{PE}|\varphi\rangle|^2}{p_0}}$$
$$= \sqrt{1 - \frac{p}{p_0}} \leq \sqrt{\frac{17\pi^2 \||w_{\mathcal{A}}\rangle\|}{16Tp}}. \qquad \square$$

3.4 Quantum walks and electrical flow

We now discuss how a quantum walk search algorithm works in the electric network framework from [Bel13] and how it intuitively connects to electrical networks. Later constructions, such as the ones in [AGJK20] and [AGJ20] that were modified to not only detect, but also find, are in essence similar.

In this thesis, the intuition of what a quantum walk algorithm entails should not be an accelerated random walk (as in [AS19]). Instead, our quantum walk algorithms are phase estimation algorithms, whose parameters are initialised and whose complexity is analysed through random walks and electrical networks. Although there already exists a relationship between the analysis of random walks and electrical networks, see for example [LP16], we will see that this relationship is even more explicit when it comes to the electrical network framework. The initialisation, and consequently the analysis, of the phase estimation algorithms is by no means unique. In the remainder of this chapter, as well as in Chapter 6, we consider the initialisation when the edge labels from Definition 3.2.13 are simply the vertices themselves and where we do not consider any networks with length as in Definition 3.2.7. In Chapter 4 we consider a more general initialisation of the phase estimation algorithm involving both non-trivial edge labels and networks with length. In Section 4.4 we will discuss the differences and similiarity of these two initialisations in more detail.

Fix a network $G = (V, E, \mathsf{w})$ and vertices $s, t \in V$. We will suppose for simplicity that σ is supported on this single vertex s, and either $M = \emptyset$ or $M = \{t\}$. This graph can be given to us as part of the search problem that we are trying to solve, as is the case in the *welded trees* problem in Section 4.4, but we can also model our search problem to a graph problem, as we do in the *k-distinctness* problem in Chapter 5. We make a slight modification to our graph, which is common trick in quantum walk algorithms to

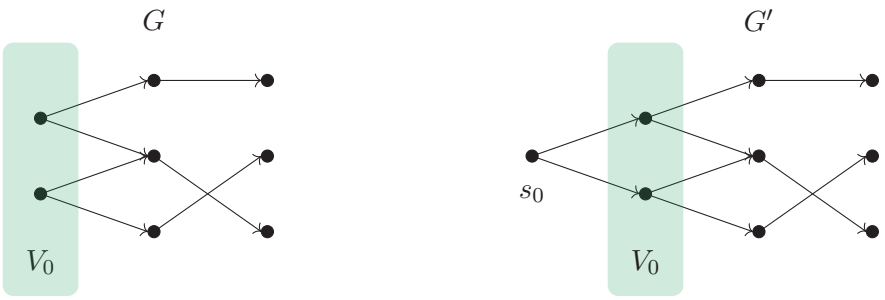


Figure 3.2: Example of a graph G with $V_0 \subseteq V(G)$ and the induced graph G' that is obtained from G by adding a new vertex s_0. This new vertex is connected to every vertex $u \in V_0$ via an edge of weight $\mathsf{w}_0\sigma_u$.

improve the final complexity of our phase estimation algorithm. Let G' be the graph G with a single extra vertex s_0, connected to s via an edge with weight w_0. We consider the direction (s_0, s) to be part of the directed edge set $\overrightarrow{E}(G')$. This modification is shown in Figure 3.2 for when σ is supported on a set $V_0 \subseteq V(G)$. This modification only allows for a slightly improved final complexity, but to gain a conceptual understanding of the algorithm, one can also ignore this modification and simply read s_0 to be s.

3.4.1 Initialising the phase estimation algorithm

To initialise our phase estimation algorithm as in Definition 3.3.1, we start by defining our Hilbert space, which we choose to be the *edge space* of our modified network G':

$$\mathcal{H} = \operatorname{span}\{|u, v\rangle : (u, v) \in E(G')\}. \tag{3.22}$$

Note that each edge $(u, v) \in E(G')$ "appears twice" in $\mathcal{H}$: both as $|u, v\rangle$ and $|v, u\rangle$, which are orthogonal states in $\mathcal{H}$. This also means that intuitively our quantum walk will "walk on edges", where as a classical random walk on a graph usually takes place on its vertices.

For each vertex $u \in V(G')$, we define the normalised *star state* of $u \in V(G')$ with respect to the graph G' as

$$\begin{aligned} |\psi_\star^{G'}(u)\rangle &:= \frac{1}{\sqrt{\mathsf{w}_u^{G'}}} \left(\sum_{v \in \Gamma_{G'}^+(u)} \sqrt{\mathsf{w}_{u,v}} |u, v\rangle - \sum_{v \in \Gamma_{G'}^-(u)} \sqrt{\mathsf{w}_{u,v}} |u, v\rangle \right) \\ &= \frac{1}{\sqrt{\mathsf{w}_u^{G'}}} \sum_{v \in \Gamma_{G'}(u)} (-1)^{\Delta_{u,v}} \sqrt{\mathsf{w}_{u,v}} |u, v\rangle. \end{aligned} \tag{3.23}$$

Because the star states are taken with respect to the graph G', we must not forget that $s_0 \in \Gamma_{G'}^-(s)$. Here for any $(u, v) \in E(G')$, the quantity $\Delta_{u,v}$ is equal to 0 if $(u, v) \in \overrightarrow{E}(G')$ and 1 if $(v, u) \in \overrightarrow{E}(G')$, where we recall that $(s_0, s) \in \overrightarrow{E}(G')$. The construction of these star states is consistent with the one from Definition 3.2.13, by viewing the vertices as labels, meaning $f_u(v) = v$ for every edge $(u, v) \in E(G')$. These star states are used to construct the set of vectors $\Psi_{\mathcal{A}} \subset \mathcal{H}$ as follows:

$$\Psi_{\mathcal{A}} := \{|\psi_\star^{G'}(u)\rangle : u \in V(G') \setminus (\{s_0\} \cup M)\}. \tag{3.24}$$

Its span, denoted by $\mathcal{A}$, then forms the *star space* of $\mathcal{H}$. The other subspace is constructed from

$$\Psi_{\mathcal{B}} := \{|u, v\rangle - |v, u\rangle : (u, v) \in E(G')\}. \tag{3.25}$$

Its span, denoted by $\mathcal{B}$, forms the *antisymmetric subspace* of $\mathcal{H}$. Under these choices of $\mathcal{H}, \Psi_{\mathcal{A}}$ and $\Psi_{\mathcal{B}}$, the unitary $U_{\mathcal{AB}}$ is now known as the *quantum walk operator*

$$U_{\mathcal{AB}} = (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I) = -\mathsf{SWAP}\left(2\sum_{u\in V(G')\setminus\{s_0,t\}} |\psi_\star^{G'}(u)\rangle\langle\psi_\star^{G'}(u)| - I\right). \quad (3.26)$$

Here SWAP acts as $\mathsf{SWAP}|u,v\rangle = |v,u\rangle$ for any $|u,v\rangle \in \mathcal{H}$.

The last parameter according to Definition 3.3.1 is the initial state $|\psi_0\rangle \in \mathcal{B}^\perp$ on which we will run the phase estimation of $U_{\mathcal{AB}}$. By our choice of $\mathcal{B}$, the space $\mathcal{B}^\perp$ forms the *symmetric subspace* of $\mathcal{H}$ and is spanned by the following set of states:

$$\{|u,v\rangle + |v,u\rangle : (u,v) \in E(G')\}. \quad (3.27)$$

To construct $|\psi_0\rangle$, we take the star state $|\psi_\star(s_0)\rangle$, which can be seen as a quantum sample of the initial distribution σ (which in our case is only supported on s), and take its its normalised projection onto $\mathcal{B}^\perp$:

$$|\psi_0\rangle := \sqrt{2}(I - \Pi_{\mathcal{B}})|\psi_\star(s_0)\rangle = \frac{1}{\sqrt{2}}\left(|s_0,s\rangle + |s,s_0\rangle\right). \quad (3.28)$$

3.4.2 Detecting a marked vertex

Having initialised our phase estimation algorithm, we can now run it and use Theorem 3.3.8 to distinguish whether the positive or negative condition holds. By linking the existence of a marked vertex to these conditions, we end up with a quantum walk algorithm that detects the existence of a marked vertex.

The positive condition: $M = \{t\}$

Suppose $M = \{t\}$, how do we use this fact to construct a positive witness as in Definition 3.3.5. We may assume that there is a path from s to t in G, otherwise a random walk from s will never find t. This means that there exists a unit s-t flow on G. For every such s-t flow θ on G we can create a (normalised) *flow state*, which is a quantum representation of the flow living in $\mathcal{B}^\perp$ (see (3.27)):

$$|\theta\rangle := \frac{1}{\sqrt{2\mathcal{E}(\theta)}} \sum_{(u,v)\in\overrightarrow{E}(G)} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}\left(|u,v\rangle + |v,u\rangle\right). \quad (3.29)$$

It might seem slightly counterintuitive that the flow state lives in the symmetric subspace, since we saw in Section 3.2.2 that the s-t flow θ as in Definition 3.2.3 could be interpreted as a vector θ in the symmetric subspace of $\mathbb{C}^{\overrightarrow{E}(G)}$, satisfying $\theta_{u,v} = \theta_{v,u}$. By Definition 3.3.5 however, we know that a 0-positive witness lies in both $\mathcal{A}^\perp$ and $\mathcal{B}^\perp$, the latter of which we have now guaranteed. To obtain a state $\mathcal{A}^\perp$, we are interested in an s_0-t flow θ' on G', which we can construct from s-t flow θ by sending one unit of flow along the edge $(s_0,s) \in \overrightarrow{E}(G')$ (with $\mathsf{w}_{s_0,s} = w_0$). By (3.29), its corresponding flow state then becomes

$$\begin{aligned}|\theta'\rangle &:= \frac{1}{\sqrt{2\left(\mathcal{E}(\theta')\right)}} \sum_{(u,v)\in\overrightarrow{E}(G')} \frac{\theta'_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}\left(|u,v\rangle + |v,u\rangle\right) \\ &= \frac{1}{\sqrt{2\left(\mathcal{E}(\theta) + 1/\mathsf{w}_0\right)}}\left(\sum_{(u,v)\in\overrightarrow{E}(G)} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}\left(|u,v\rangle + |v,u\rangle\right) + \frac{1}{\mathsf{w}_0}\left(|s_0,s\rangle + |s,s_0\rangle\right)\right).\end{aligned} \quad (3.30)$$

To ensure orthogonality with $\mathcal{A}^\perp$, note that this is precisely the space orthogonal to all star states of vertices in $V(G') \setminus \{s_0, t\}$. By Kirchhoff's Law (see Definition 3.2.5), we know that each flow state $|\theta'\rangle$ must be orthogonal to such star states: if the s_0-t flow θ' goes through a vertex $u \in V(G') \setminus \{s_0, t\}$, it is supported on 2 of the edges adjacent to u, one contributing -1 because it goes into u, and the other $+1$ because it comes out of u. More formally, we have for any vertex $u \in V(G') \setminus \{s_0, t\}$ that

$$\begin{aligned}\langle \psi_\star^{G'}(u)|\theta'\rangle &\propto \sum_{v\in\Gamma_{G'}(u)} (-1)^{\Delta_{u,v}} \sqrt{\mathsf{w}_{u,v}}\langle u,v| \sum_{(u,v)\in\overrightarrow{E}(G')} \frac{\theta'_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}\left(|u,v\rangle + |v,u\rangle\right) \qquad \text{see (3.23)}\\ &= \sum_{v\in\Gamma^+_{G'}(u)} (-1)^{\Delta_{u,v}}\theta'_{u,v} + \sum_{v\in\Gamma^+_{G'}(u)} (-1)^{\Delta_{u,v}}\theta'_{v,u} = \sum_{v\in\Gamma_{G'}(u)} \theta'_{u,v} = 0, \qquad (3.31)\end{aligned}$$

where we used the fact that $(-1)^{\Delta_{u,v}} = 1$ when $v \in \Gamma^+_{G'}(u)$ and (-1) if $v \in \Gamma^-_{G'}(u)$, that $\theta'_{v,u} = -\theta'_{u,v}$, and that θ' is an s_0-t flow on G', meaning it is conserved at every $u \in V(G') \setminus \{s_0, t\}$. It follows similarly that $|\theta'\rangle$ has non-zero overlap with our initial state:

$$\begin{aligned}\langle\psi_0|\theta\rangle &= \frac{1}{\sqrt{2\left(\mathcal{E}(\theta) + 1/\mathsf{w}_0\right)}}\frac{1}{\sqrt{2}}\left(\langle s_0, s| + \langle s, s_0|\right) \sum_{(u,v)\in\overrightarrow{E}(G')} \frac{\theta'_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}(|u,v\rangle + |v,u\rangle)\\ &= \frac{1}{2\sqrt{\mathsf{w}_0\mathcal{E}(\theta)+1}}\left(\theta'_{s_0,s} - \theta'_{s,s_0}\right) = \frac{1}{\sqrt{\mathsf{w}_0\mathcal{E}(\theta)+1}}.\end{aligned} \qquad (3.32)$$

We have hence derived that $|\theta'\rangle$ is a valid 0-positive witness for every s-t unit flow θ. But what would be the optimal positive witness, as defined in Definition 3.3.9. For now, let us assume for simplicity that the optimal positive witness is equal to $|\theta'\rangle$ for some s-t unit flow θ (we will see in the proof of Corollary 3.4.2 that this must indeed be the case). Then by Definition 3.3.9 and (3.32), the optimal positive witness is given by

$$|w\rangle = \underset{\sqrt{\mathsf{w}_0\mathcal{E}(\theta)+1}|\theta'\rangle}{\arg\min}\left\{\left\|\sqrt{\mathsf{w}_0\mathcal{E}(\theta)+1}|\theta'\rangle\right\| : \theta \text{ is a } s\text{-}t \text{ unit flow}\right\}. \qquad (3.33)$$

This means that the optimal positive witness is given by

$$|w\rangle = \sqrt{\mathsf{w}_0\mathcal{R}_{s,t}+1}|\theta'\rangle, \qquad (3.34)$$

where θ is the s-t electrical flow (see Definition 3.2.3). To ensure that $\frac{|\langle w|\psi_0\rangle|^2}{\||w\rangle\|^2}$ is at least a constant as required by Theorem 3.3.8, we set $\mathsf{w}_0 = \frac{1}{\mathcal{R}_{s,t}}$. This is valid choice of w_0, as it yields

$$\mathsf{w}_0\mathcal{R}_{s,t} + 1 = 2, \qquad (3.35)$$

meaning that it satisfies

$$\frac{|\langle w|\psi_0\rangle|^2}{\||w\rangle\|^2} = \frac{1}{2}. \qquad (3.36)$$

The negative condition: $M = \emptyset$

On the other hand, suppose $M = \emptyset$. Contrary to the positive case, this means that the star state of t is now one of the states spanning $\mathcal{A}$. We can make use of this to decompose our initial state $|\psi_0\rangle$ into a component $|w_\mathcal{A}\rangle \in \mathcal{A}$ and a component $|w_\mathcal{B}\rangle \in \mathcal{B}$ to create a negative witness as in Definition 3.3.2. First, observe that any two distinct star states $|\psi_\star^{G'}(u)\rangle, |\psi_\star^{G'}(v)\rangle$ are orthogonal, since the former is supported on states of the form $|u,w\rangle$ with $w \in \Gamma_{G'}(u)$, where as the latter is supported on states of the form $|v,w\rangle$ where

$w \in \Gamma_{G'}(v)$. Hence, by taking the sum over all star states of vertices in $V(G)$, which lies in $\mathcal{A}$ since $M = \emptyset$, no interference occurs and we obtain a weighted sum over all edges in $E(G')$:

$$\sum_{u \in V(G)} \sqrt{\mathsf{w}_u^{G'}} |\psi_\star^{G'}(u)\rangle = -\sqrt{\mathsf{w}_0}|s, s_0\rangle + \sum_{(u,v)\in E(G)} (-1)^{\Delta_{u,v}} \sqrt{\mathsf{w}_{u,v}}|u, v\rangle. \tag{3.37}$$

Note that the above equation almost lies in $\mathcal{B}$ (see (3.25)), as

$$\sum_{(u,v)\in E(G)} (-1)^{\Delta_{u,v}} \sqrt{\mathsf{w}_{u,v}}|u, v\rangle = \sum_{(u,v)\in \vec{E}(G)} \sqrt{\mathsf{w}_{u,v}}\,(|u, v\rangle - |v, u\rangle) \in \mathcal{B}. \tag{3.38}$$

From these two observations, we derive the following choice for our negative witness. Let

$$\begin{aligned} |w_{\mathcal{A}}\rangle &:= -\sum_{u\in V(G)} \frac{\sqrt{2\mathsf{w}_u^{G'}}}{\sqrt{\mathsf{w}_0}} |\psi_\star^{G'}(u)\rangle, \\ |w_{\mathcal{B}}\rangle &:= \frac{1}{\sqrt{2}}\,(|s_0, s\rangle - |s, s_0\rangle) + \sum_{(u,v)\in \vec{E}(G)} \frac{\sqrt{2\mathsf{w}_{u,v}}}{\sqrt{\mathsf{w}_0}}\,(|u, v\rangle - |v, u\rangle)\,. \end{aligned} \tag{3.39}$$

Then by (3.37) and (3.38) we find that

$$|w_{\mathcal{A}}\rangle + |w_{\mathcal{B}}\rangle = \sqrt{2}|s, s_0\rangle + \frac{1}{\sqrt{2}}\,(|s_0, s\rangle - |s, s_0\rangle) = \frac{1}{\sqrt{2}}\,(|s_0, s\rangle + |s, s_0\rangle) = |\psi_0\rangle. \tag{3.40}$$

The complexity

Having satisfied all the requirements in Theorem 3.3.8, we can now use it to analyse the complexity of our phase estimation algorithm that detects whether the vertex t is marked:

Corollary 3.4.1. *Fix a network $G = (V, E, \mathsf{w})$ as in Definition 3.2.1 with vertices $s, t \in V$. Let $U_{\mathcal{AB}}$ be the quantum walk operator as defined in (3.26). Then by performing $T = 2\sqrt{8}\pi^4\sqrt{\mathcal{R}_{s,t}\mathcal{W} + 2}$ steps of phase estimation on the initial state $|\psi_0\rangle$ as defined in (3.28) with the operator $U_{\mathcal{AB}}$, the phase estimation algorithm distinguishes whether $M = \{t\}$ or $M = \emptyset$ with bounded error.*

Before we prove this statement, we note that it is straightforward to extend this approach to general σ and a marked set M that contains possibly more vertices than just t. Our general multidimensional quantum walk framework in Section 4.3 will be able to deal with these more general cases, and it will also take into account the cost of implementing $U_{\mathcal{AB}}$, as well as generating the initial state $|\psi_0\rangle$.

Proof. Since we already constructed our positive witness in (3.34) and our negative witness in (3.39), we only have to compute the quantities c_+ and C_- from Theorem 3.3.8. We have already seen in (3.36) that

$$\frac{|\langle w|\psi_0\rangle|^2}{\||w\rangle\|^2} = \frac{1}{2},$$

meaning $c_+ = 2$. Since we also chose $\mathsf{w}_0 = \frac{1}{\mathcal{R}_{s,t}}$, we find by (3.37) that

$$\begin{aligned} \||w_{\mathcal{A}}\rangle\|^2 &= 2\mathcal{R}_{s,t} \left\| -\sqrt{\mathsf{w}_0}|s, s_0\rangle + \sum_{(u,v)\in E(G)} (-1)^{\Delta_{u,v}} \sqrt{\mathsf{w}_{u,v}}|u, v\rangle \right\|^2 \\ &= 2\mathcal{R}_{s,t} \left(\sum_{e\in E(G)} \mathsf{w}_e + \mathsf{w}_0 \right) = \mathcal{R}_{s,t}\mathcal{W} + 2, \end{aligned}$$

meaning $C_- = \mathcal{R}_{s,t}\mathcal{W} + 2$ suffices. □

When applying Corollary 3.4.1, or even our more general multidimensional quantum walk framework in Section 4.3, it is sufficient to provide an upper bound on the quantity $\mathcal{R}_{s,t}$ (and $\mathcal{W}$). Instead of trying to construct and upper bound the energy of the electrical s-t flow, we can simply construct any unit s-t flow and upper bound its energy, as this will always be an upper bound on $\mathcal{R}_{s,t}$. A nice way to interpret this is that the quantity $\mathcal{R}_{s,t}\mathcal{W}$ is equal to the *commute time* from s to t – the expected number of steps a random walker starting from s needs to reach t, and then return to s. For a discussion of how to interpret this quantity in the case of more general σ and M, see [AGJ20].

3.4.3 Approximating the electrical flow

We can use the same initialisation of the parameters of our phase estimation algorithm, to invoke Lemma 3.3.11 and exhibit an algorithm that approximates (a normalised version) of the optimal positive witness, which in our case (see (3.34)) is the flow state $|\theta'\rangle$, constructed from the s-t electrical flow θ. In fact, given an ϵ-approximation of $|\theta'\rangle$, we show that with high probability we can obtain an ϵ-approximation of $|\theta\rangle$

The only unknown quantity yet in Lemma 3.3.11 is $|w_{\mathcal{A}}\rangle$ (not to be confused with $|w_{\mathcal{A}}\rangle$ from the negative witness in Definition 3.3.2). This will be equal to the (unnormalised) state associated with the induced potential vector p corresponding to the s-t electrical flow θ (with the convention that $\mathsf{p}_t = 0$):

$$|\mathsf{p}\rangle := \sqrt{\frac{2}{\mathcal{R}_{s,t}}} \sum_{u \in V(G)\setminus\{s\}} \mathsf{p}_u \sqrt{\mathsf{w}_u} |\psi_\star^G(u)\rangle. \tag{3.41}$$

In [Pid19, AP22], this potential state $|\mathsf{p}\rangle$ is used to apply Lemma 3.3.11 and show that by running phase estimation on the quantum walk operator $U_{\mathcal{A}B}$, we can obtain a close approximation to the flow state $|\theta\rangle$. The precision required in this phase estimation algorithm scales with a quantity that [AP22] define as the escape time ET_s:

$$\mathsf{ET}_s := \frac{1}{\mathcal{R}_{s,t}} \sum_{u \in V(G)} \mathsf{p}_u^2 \mathsf{w}_u. \tag{3.42}$$

This operational meaning of this quantity, is that it captures the expected time where a random walk leaves s for the final time, before it arrives at t.

Corollary 3.4.2. *Fix a network $G = (V, E, \mathsf{w})$ as in Definition 3.2.1 with vertices $s, t \in V$. Let $U_{\mathcal{A}B}$ be the quantum walk operator as defined in (3.26). Let θ be the s-t electrical flow on G with corresponding flow state $|\theta\rangle$ as defined in (3.29). Then by performing $T = \frac{17\pi^2}{16\sqrt{2}\epsilon^2}\sqrt{\mathsf{ET}_s + 1}$ steps of phase estimation on the initial state $|\psi_0\rangle$ as defined in (3.28) with the operator $U_{\mathcal{A}B}$, the phase estimation algorithm outputs "0" with bounded error, leaving a state $|\widetilde{\theta}\rangle$ satisfying*

$$\frac{1}{2}\left\| |\widetilde{\theta}\rangle\langle\widetilde{\theta}| - |\theta\rangle\langle\theta| \right\|_1 \leq \epsilon.$$

Proof. We will write θ for the s-t electrical flow on G and θ' for the s_0-t electrical flow on G'. First, recall our choice of positive witness $|w\rangle = \sqrt{2}|\theta'\rangle$ from (3.34). We claimed that this was in fact the optimal positive witness, but we have not yet proven this claim. The optimality of $|w\rangle$ was not needed to use the phase estimation algorithm to detect a marked vertex, but it will be necessary if we want to use Lemma 3.3.11. Since θ' is the s_0-t electrical flow on G', we know by Ohm's Law (see Definition 3.2.6) that there exists a potential p', with $\mathsf{p}'_t = 0$.

$$\mathsf{p}'_{s_0} = \mathcal{R}_{s_0,t} = \mathcal{R}_{s,t} + \frac{1}{\mathsf{w}_0} = 2\mathcal{R}_{s,t},$$

and satisfying $\mathsf{p}_u - \mathsf{p}_v = \frac{\theta_{u,v}}{\mathsf{w}_{u,v}}$ for each edge $(u,v) \in E(G)$. By (3.41), the corresponding potential state of p' is

$$|\mathsf{p}'\rangle := \sqrt{\frac{2}{\mathcal{R}_{s_0,t}}} \sum_{u \in V(G')\setminus\{s_0\}} \mathsf{p}'_u \sqrt{\mathsf{w}_u^{G'}} |\psi_\star^{G'}(u)\rangle = \sqrt{\frac{1}{\mathcal{R}_{s,t}}} \sum_{u \in V(G)} \mathsf{p}'_u \sqrt{\mathsf{w}_u^{G'}} |\psi_\star^{G'}(u)\rangle.$$

This potential allows us to decompose the flow state $|f'\rangle$ from (3.30) as

$$\begin{aligned}
|\theta'\rangle &= \frac{1}{2\sqrt{\mathcal{R}_{s,t}}} \sum_{(u,v)\in\vec{E}(G')} \frac{\theta'_{u,v}}{\sqrt{\mathsf{w}_{u,v}}} (|u,v\rangle + |v,u\rangle) \\
&= \frac{1}{2\sqrt{\mathcal{R}_{s,t}}} \sum_{(u,v)\in\vec{E}(G')} \left(\sqrt{\mathsf{w}_{u,v}}(\mathsf{p}'_u - \mathsf{p}'_v)|u,v\rangle + (\mathsf{p}'_u - \mathsf{p}'_v)\sqrt{\mathsf{w}_{u,v}}|v,u\rangle\right) \\
&= \frac{1}{2\sqrt{\mathcal{R}_{s,t}}} (I + \mathsf{SWAP}) \sum_{u\in V(G')} \mathsf{p}'_u \sum_{v\in\Gamma_{G'}(u)} (-1)^{\Delta_{u,v}} \sqrt{\mathsf{w}_{u,v}} |u,v\rangle \\
&= (I - \Pi_{\mathcal{B}}) \frac{1}{\sqrt{\mathcal{R}_{s,t}}} \sum_{u\in V(G')} \mathsf{p}'_u \sqrt{\mathsf{w}_u^{G'}} |\psi_\star(u)\rangle \qquad \text{see (3.23)} \\
&= (I - \Pi_{\mathcal{B}})|\mathsf{p}'\rangle + (I - \Pi_{\mathcal{B}}) \frac{1}{\sqrt{\mathcal{R}_{s,t}}} \mathsf{p}'_{s_0} \sqrt{\mathsf{w}_{s_0}} |\psi_\star(s_0)\rangle \\
&= (I - \Pi_{\mathcal{B}})|\mathsf{p}'\rangle + (I - \Pi_{\mathcal{B}}) \frac{1}{\sqrt{\mathcal{R}_{s,t}}} 2\sqrt{\mathcal{R}_{s,t}} |\psi_\star(s_0)\rangle = (I - \Pi_{\mathcal{B}})|\mathsf{p}'\rangle + \sqrt{2}|\psi_0\rangle. \qquad \text{see (3.28)}
\end{aligned} \tag{3.43}$$

This can be rearranged to

$$|\psi_0\rangle = \frac{1}{\sqrt{2}}|\theta'\rangle - (I - \Pi_{\mathcal{B}}) \frac{1}{\sqrt{2}} |\mathsf{p}'\rangle.$$

Since $\mathsf{p}'_t = 0$, we immediately have by its definition in (3.41) that $|\mathsf{p}'\rangle \in \mathcal{A}$, meaning we have shown that our choice of $|w\rangle$ was in fact the optimal positive witness due to Fact 3.3.10. Hence, by applying Lemma 3.3.11, with $|w_{\mathcal{A}}\rangle = \frac{1}{\sqrt{2}}|\mathsf{p}'\rangle$, we find that the resulting state after running phase estimation on the quantum walk operator $U_{\mathcal{AB}}$ with initial state $|\psi_0\rangle$ is approximately the s_0-t electrical flow state $|\theta'\rangle$. To turn this into $|\theta\rangle$, observe that due to (3.30) we can decompose

$$|\theta'\rangle = \frac{1}{2\sqrt{\mathcal{R}_{s,t}}} \left(\sqrt{2\mathcal{R}_{s,t}}\,(|s_0,s\rangle + |s,s_0\rangle) + \sqrt{2\mathcal{R}_{s,t}}|\theta\rangle\right). \tag{3.44}$$

By (3.35) we therefore know that if we postselect on our edges not containing the vertex s_0, we are left with $|\theta\rangle$ with success probability $1/2$.

Lastly, by Ohm's Law (see Definition 3.2.6) it is straightforward to see that the potentials p and p' coincide on $V(G)$, meaning we can relate $\||\mathsf{p}'\rangle\|$ to ET_s (see (3.42)) as

$$\left\||\mathsf{p}'\rangle\right\|^2 = \frac{1}{\mathcal{R}_{s,t}} \sum_{u\in V(G)} \mathsf{p}_u^2 \mathsf{w}_u^{G'} = \frac{1}{\mathcal{R}_{s,t}} \sum_{u\in V(G)} \mathsf{p}_u^2 \mathsf{w}_u + \mathcal{R}_{s,t}\mathsf{w}_0 = \mathsf{ET}_s + 1. \qquad \square$$

In this section, we have shown what we mean with quantum walks in the electrical network framework, namely phase estimation algorithms of the specific form from Theorem 3.3.8, initialised with objects originating from random walks and electrical networks. Table 3.1 summarises this initialisation.

	Phase estimation	Quantum walk
Definition 3.3.1	$\mathcal{H}$	$\mathrm{span}\{\|u,v\rangle : (u,v) \in E(G')\}$
	$\|\psi_0\rangle$	$\frac{1}{\sqrt{2}}(\|s_0,s\rangle + \|s,s_0\rangle)$
	$\Psi_{\mathcal{A}}$	$\{\|\psi_\star(u)\rangle : u \in V(G')\setminus(\{s_0\}\cup M)\}$
	$\Psi_{\mathcal{B}}$	$\{\|u,v\rangle - \|v,u\rangle : (u,v) \in E(G')\}$
Theorem 3.3.8	C_-	$\mathcal{R}_{s,t}\mathcal{W} + 2$
	$\|w\rangle$	$\sqrt{\frac{\mathcal{E}(\theta')}{\mathcal{R}_{s,t}}}\|\theta'\rangle$ for any unit s_0-t flow θ'
	$\|w_{\mathcal{A}}\rangle$	$-\sum_{u\in V(G)}\sqrt{2\mathsf{w}_u^{G'}\mathcal{R}_{s,t}}\|\psi_\star^{G'}(u)\rangle$
	$\|w_{\mathcal{B}}\rangle$	$\frac{1}{\sqrt{2}}(\|s_0,s\rangle - \|s,s_0\rangle)$ $+\sum_{(u,v)\in\vec{E}(G)}\sqrt{2\mathsf{w}_{u,v}\mathcal{R}_{s,t}}(\|u,v\rangle - \|v,u\rangle)$
Lemma 3.3.11	$\|w\rangle$	$\sqrt{2}\|\theta'\rangle$ with θ' being the s_0-t electrical flow
	$\|w_{\mathcal{A}}\rangle$	$\frac{1}{\sqrt{2}}\|\mathsf{p}'\rangle$, with p' being the potential corresponding to the s_0-t electrical flow

Table 3.1: A summary of how we initialise the phase estimation parameters in Section 3.4, as well as the parameters in Theorem 3.3.8 and Lemma 3.3.11, by using states and spaces whose construction is inspired by objects from random walks and electrical networks.

3.4.4 Example graph

We return to our example graph from Figure 3.1, which is shown again here in Figure 3.3 for convenience. The directed edges and weight assignments in the network G give rise to the following star states (see (3.23)) for each of the four vertices:

$$|\psi_\star(s)\rangle = |s,x\rangle, \qquad |\psi_\star(x)\rangle = \sqrt{\frac{2}{3}}\left(-|x,s\rangle + \frac{1}{2}|x,y\rangle + \frac{1}{2}|x,t\rangle\right),$$
$$|\psi_\star(y)\rangle = \sqrt{2}\left(-\frac{1}{2}|y,x\rangle + \frac{1}{2}|y,t\rangle\right), \qquad |\psi_\star(t)\rangle = \sqrt{2}\left(-\frac{1}{2}|t,x\rangle - \frac{1}{2}|t,y\rangle\right).$$

The flow state $|\theta\rangle$ (see (3.29)), corresponding to the s-t electrical flow θ visualised in Figure 3.3, is given by

$$|\theta\rangle = \sqrt{\frac{3}{22}}\left(|s,x\rangle + |x,s\rangle + \frac{2}{3}|x,y\rangle + \frac{2}{3}|y,x\rangle + \frac{4}{3}|x,t\rangle + \frac{4}{3}|t,x\rangle + \frac{4}{3}|y,t\rangle + \frac{2}{3}|t,y\rangle\right).$$

It is straightforward to verify that $|\theta\rangle$ is orthogonal to the star states $|\psi_\star(x)\rangle$ and $|\psi_\star(y)\rangle$ corresponding to the vertices x and y, respectively. The resulting potential state $|\mathsf{p}\rangle$ (see (3.41)) corresponding to the potential visualised in Figure 3.3 is

$$|\mathsf{p}\rangle = -\frac{8}{3}|x,s\rangle + \frac{4}{3}|x,y\rangle + \frac{4}{3}|x,t\rangle - \frac{2}{3}|y,x\rangle + \frac{2}{3}|y,t\rangle. \tag{3.45}$$

3.5 Model of computation and quantum subroutines

Our quantum walks work in the (fully quantum) QRAM model, which we now describe. By QRAM, we mean *quantum* memory, storing an arbitrary quantum state, to which we

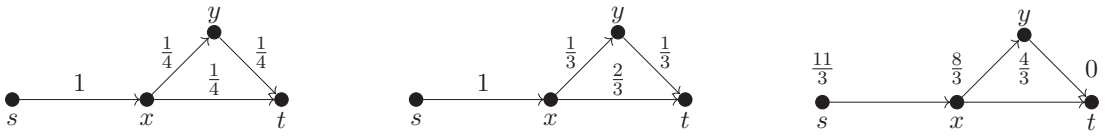


Figure 3.3: Graph G with its s-t electrical flow θ and corresponding potential p at each vertex.

can apply random access gates. By this, we mean we can implement, for $i \in [n]$, $b \in \{0,1\}$, and $x \in \{0,1\}^n$, a random access read:

$$\text{READ} : |i\rangle|b\rangle|x\rangle \mapsto |i\rangle|b \oplus x_i\rangle|x\rangle,$$

or a random access write:

$$\text{WRITE} : |i\rangle|b\rangle|x\rangle \mapsto |i\rangle|b\rangle|x_1, \ldots, x_{i-1}, x_i \oplus b, x_{i+1}, \ldots, x_n\rangle,$$

on any superposition. By applying READ · WRITE · READ, we can implement a controlled SWAP:

$$\text{READ} \cdot \text{WRITE} \cdot \text{READ}(|i\rangle|b\rangle|x\rangle) = |i\rangle|x_i\rangle|x_1, \ldots, x_{i-1}, b, x_{i+1}, \ldots, x_n\rangle.$$

Aside from these operations, we count the number of elementary gates, by which we mean arbitrary unitaries that act on $O(1)$ qubits.

We will be interested in running different iterations of a subroutine on the different branches of a superposition, for which we use the concept of a quantum subroutine. We note that Definition 3.5.1 is *not* the most general definition, but it is sufficient for our purposes.

Definition 3.5.1 (Quantum Subroutine)**.** *A* quantum subroutine *is a sequence of unitaries $U_0, \ldots, U_{\mathsf{T}_{\max}-1}$ on $H_{\mathcal{Z}} = \operatorname{span}\{|z\rangle : z \in \mathcal{Z}\}$ for some finite set $\mathcal{Z}$. For $X, Y \subseteq \mathcal{Z}$, we say the subroutine computes an injective function $f : X \to Y$ in times $\{\mathsf{T}_x \leq \mathsf{T}_{\max}\}_{x \in X}$ with errors $\{\epsilon_x\}_{x \in X}$ if:*

1. *The map $\sum_{t=0}^{\mathsf{T}_{\max}-1} |t\rangle\langle t| \otimes U_t$ can be implemented in $\mathsf{polylog}(\mathsf{T}_{\max})$ complexity.*
2. *For all $x \in X$, $\| |f(x)\rangle - U_{\mathsf{T}_x - 1} \ldots U_0(|x\rangle) \|^2 \leq \epsilon_x$.*
3. *The maps $x \mapsto \mathsf{T}_x$ and $y \mapsto \mathsf{T}_{f^{-1}(y)}$ can both be implemented in $\mathsf{polylog}(\mathsf{T}_{\max})$ complexity.*
4. *There exists a decomposition $\mathcal{Z} = \bigcup_{x \in X} \mathcal{Z}_x$ such that $x, f(x) \in \mathcal{Z}_x$, and for every $t \in [\mathsf{T}_{\max} - 1]_0$, $U_t \ldots U_0 |x\rangle \in \operatorname{span}\{|z\rangle : z \in \mathcal{Z}_x\}$.*

While not all of our assumptions are general, they are reasonable in our setting. Item 1 is standard in subroutines that will be run in superposition (see e.g. [Amb10b]), and is reasonable, for example, in settings where the algorithm is sufficiently structured to compute U_t from a standard gate set on the fly, which we formalise in Lemma 3.5.2 below (see also the discussion in [CJOP20, Section 2.2]).

Item 3 is not always necessary, but it is often true, and simplifies things considerably. It means, in particular, that one can decide, based on the input, how many steps of the algorithm should be applied, and then, based on the output, uncompute this information.

Item 4 is not a standard assumption, but it is also not unreasonable. For example, if $X = X' \times \{0\}$ and $f(x, 0) = (x, g(x))$ for some function g, the algorithm may simply use x as a control, and so its state always encodes x, and therefore remains orthogonal for different x.

Lemma 3.5.2. *Call unitaries $U_0, \dots, U_{\mathsf{T}_{\max}-1}$ on $\mathcal{H}$ a* uniform quantum algorithm *if there exists $\ell = \mathsf{polylog}(\mathsf{T}_{\max})$, unitaries $W_1, \dots, W_\ell$, and maps $g : [\mathsf{T}_{\max} - 1]_0 \to [\ell]$ and $g' : [\mathsf{T}_{\max} - 1]_0 \to 2^{[\log \dim H]}$ such that:*

1. *For each $j \in [\ell]$, W_j can be implemented by $\mathsf{polylog}(\mathsf{T}_{\max})$ gates from some implicit gate set (and therefore acts on $m = \mathsf{polylog}(\mathsf{T}_{\max})$ qubits).*
2. *g and g' can be computed in $\mathsf{polylog}(\mathsf{T}_{\max})$ complexity.*
3. *For all $t \in [\mathsf{T}_{\max} - 1]_0$, $U_t = W_{g(t)}(g'(t))$, where $W_\ell(S)$ denotes W_ℓ applied to the qubits specified by S.*

Then $\sum_{t=0}^{\mathsf{T}_{\max}-1} |t\rangle\langle t| \otimes U_t$ can be implemented in $\mathsf{polylog}(\mathsf{T}_{\max})$ gates.

Proof. We describe how to implement $\sum_{t=0}^{\mathsf{T}_{\max}-1} |t\rangle\langle t| \otimes U_t$ on $|t\rangle|z\rangle$ for $|z\rangle \in H$. Append registers $|0\rangle_A|0\rangle_{A'} \in \mathrm{span}\{|j, S\rangle : j \in [\ell]_0, S \in \mathcal{S}\}$, where $\mathcal{S}$ is the set of subsets of $[\log \dim H]$ of size at most m. Compute $g(t)$ and $g'(t)$ to get $|t\rangle|z\rangle|0\rangle_A|0\rangle_{A'} \mapsto |t\rangle|z\rangle|g(t)\rangle_A|g'(t)\rangle_{A'}$. Controlled on $g'(t)$, we can swap the qubits acted on by U_t into the first m positions. Then we can implement $\sum_{j=1}^{\ell} |j\rangle\langle j| \otimes W_j + |0\rangle\langle 0| \otimes I$ by decomposing it into a sequence of ℓ controlled operations:

$$\prod_{j=1}^{\ell} \left(|j\rangle\langle j| \otimes W_j + (I - |j\rangle\langle j|) \otimes I\right).$$

The result follows from noticing that each of these $\ell = \mathsf{polylog}(\mathsf{T}_{\max})$ operations can be implemented with $\mathsf{polylog}(\mathsf{T}_{\max})$ controlled gates. □

Lemma 3.5.3. *Fix a constant integer c, and for $j \in [c]$, let $\mathcal{S}_j$ be a quantum subroutine on $H_j = \mathrm{span}\{|j\rangle\} \otimes H$ for some space $\mathcal{H}$ that takes time $\{\mathsf{T}_x = \mathsf{T}_j\}_{x \in X_j}$ with errors $\{\epsilon_x = \epsilon_j\}_{x \in X_j}$. Then there is a quantum algorithm that implements $\sum_{j=1}^{c} |j\rangle\langle j| \otimes \mathcal{S}_j$ in variable times $\mathsf{T}_{j,x} = O(\mathsf{T}_j)$ and errors $\epsilon_{j,x} = \epsilon_j$ for all $x \in X_j$.*

Proof. Pad each algorithm with identities so that they all have the same number, $\mathsf{T}_{\max} = \max_{j \in [c]} \mathsf{T}^{(c)}_{\max}$ of unitaries. Then for each time $t = [c\mathsf{T}_{\max} - 1]_0$, with $t = qc + r$ for $r \in [c]_0$, let $U_t = |r\rangle\langle r| \otimes U_q^{(r)} + (I - |r\rangle\langle r|) \otimes I$. □

3.5.1 Quantum data structures

We will assume we have access to a data structure that can store a set of keyed items, $S \subset \mathcal{I} \times \mathcal{K}$, for finite sets $\mathcal{K}$ and $\mathcal{I}$. For such a stored set S, we assume the following can be implemented in $\mathsf{polylog}(|\mathcal{I} \times \mathcal{K}|)$ complexity:

1. For $(i, k) \in \mathcal{I} \times \mathcal{K}$, insert (i, k) into S.
2. For $(i, k) \in S$, remove (i, k) from S.
3. For $k \in \mathcal{K}$, query the number of $i \in \mathcal{I}$ such that $(i, k) \in S$.
4. For $k \in \mathcal{K}$, return the smallest i such that $(i, k) \in S$.
5. Generate a uniform superposition over all $(i, k) \in S$.

In addition, for quantum interference to take place, we assume the data structure is coherent, meaning it depends only on S, and not on, for example, the order in which elements were added. See [BLPS22, Section 3.1] for an example of such a data structure.

Chapter 4

Multidimensional quantum walks

Upward, not Northward!

Edwin A. Abbott, Flatland: A Romance of Many Dimensions

This chapter is based on Section 3 and 4 in the paper *Multidimensional Quantum Walks, with Application to k-Distinctness* [JZ23], which is joint work with Stacey Jeffery.

While (discrete) quantum walk frameworks make it easy to design quantum algorithms, even without an in-depth knowledge of quantum computing, their primary drawback is that they can provide at most a quadratic speedup over their classical counterparts. In this chapter, we present a new framework for designing quantum walk search algorithms that overcomes this limitation, which we call the multidimensional quantum walk framework. By leveraging the novel concept of alternative neighbourhoods, this framework allows for a significantly more efficient implementation of the quantum walk operator. After introducing our framework, we demonstrate its power by applying it to the welded trees problem from [CCD+03], solving it in $O(n)$ queries and $O(n^2)$ time. This result shows that our new quantum walk framework is capable of achieving exponential speedups.

4.1 Beyond the electrical network framework

We have seen in the previous chapter that the electrical network framework from [Bel13] provides a powerful, yet intuitive recipe to construct quantum walk algorithms. In Table 3.1 we saw how the phase estimation parameters correspond to objects from random walks and electrical networks. Only by viewing the resulting quantum walk algorithm from both perspectives, can we obtain insights into how this framework can be further improved.

4.1.1 Edge composition

The first improvement to the electrical network framework comes from seeing our quantum walk as a random walk on a graph. To implement the unitary $U_{\mathcal{AB}}$, we perform a mapping that acts, for any $u \in V(G)$, as $|u,0\rangle \mapsto |\psi_\star^{G'}(u)\rangle$ (the star state of u with respect to the modified graph G'). Loosely speaking, what this usually means is that we have a labelling of the edges coming out of u, and some way of computing (u,v) from (u,i), where v is the i-th neighbour of u. If this computation costs $\mathsf{T}_{u,v}$ steps, then it takes $O(\max_{u,v} \mathsf{T}_{u,v})$ steps to implement $U_{\mathcal{AB}}$. However, in case this cost varies significantly over different u, v, we can do much better. We show how we can obtain a unitary with polylogarithmic cost, and essentially consider, in the analysis of the resulting algorithm, a quantum walk on a modified graph in which an edge (u,v) is replaced by a path of length $\mathsf{T}_{u,v}+2$, which we name *edge composition*. A similar result was already known for *learning graphs*, when a transition could be implemented with $\mathsf{T}_{u,v}$ *queries* [Bel12b]. This is an extremely useful, if not particularly surprising, feature of the framework, which we use in our application to k-distinctness in Chapter 5.

4.1.2 Alternative neighbourhoods

The second improvement comes from interpreting our quantum walk as purely a phase estimation algorithm and it is the more interesting way how we augment the electric network framework. In order to generate the star state of a vertex u, which is a superposition of the edges coming out of u, one must, in some sense, know the neighbours of u, as well as their relative weights, which is also the true for the update step in classical random walks. In certain settings, the algorithm will know that the star state for u is one of a small set of easily preparable states $\Psi_\star(u) = \{|\psi_\star^1(u)\rangle, |\psi_\star^2(u)\rangle, \dots\}$, but computing precisely which one of these is the correct state would be computationally expensive. In that case, we include all of $\Psi_\star(u)$, which we call the set of *alternative neighbourhood* of u, when constructing the spaces $\mathcal{A}$ and $\mathcal{B}$. In the case when $M = \emptyset$ in Theorem 3.3.8, the analysis is the same – by increasing $\mathcal{A}+\mathcal{B}$, we have only made the analysis easier. However, in the case $M \neq \emptyset$, the analysis has become more constrained. There are now some extra states in $\mathcal{A}+\mathcal{B}$ and constructing a positive witness from the s-t electrical flow – or any related electrical network object—may no longer be feasible.

We explain the alternative neighbourhoods technique more in-depth with examples in Section 4.2. We first remark on the unifying idea from which both these techniques follow, which incorporates both the random walk, as well as the phase estimation perspective. If we let $\{|\psi_\star(u)\rangle\}_{u\in V}$ be *any* set of states, we can make a graph G on V by letting u and v be adjacent if and only if $\langle\psi_\star(u)|\psi_\star(v)\rangle \neq 0$. Then, if this graph is bipartite, and we can reflect around the span of each state individually, we can reflect around the joint $\mathrm{span}\{|\psi_\star(u)\rangle : u \in V\}$. Quantum walk search algorithms can be seen as a special case of this, where we additionally exploit the structure of the graph to analyse the complexity of this procedure. One way of viewing alternative neighbourhoods from the graph picture, is that we extend this reasoning to the case where we have *spaces* $\{\mathrm{span}\{\Psi_\star(u)\}\}_{u\in V}$, each

of which we can efficiently reflect around, and G is now a (bipartite) graph encoding the overlap of the *spaces*, hence the qualifier *multidimensional*.

Edge composition also exploits this picture. We can define a sequence of subspaces $\{\Psi_t^{u,v}\}_{t=1}^{\mathsf{T}_{u,i}}$ that only overlap for adjacent t, and such that the subroutine computing $|v,j\rangle$ from $|u,i\rangle$ can be seen as moving through these spaces. Now the overlap graph of all these spaces will look like G, except with each edge (u,v) replaced by a path of length $\mathsf{T}_{u,v}+2$. See Figure 4.7 and Figure 4.9 for examples of such overlap graphs.

4.2 Alternative neighbourhoods

As discussed earlier, the multidimensional quantum walk framework modifies the quantum walk operator through the use of *alternative neighbourhoods*.

Definition 4.2.1 (Alternative Neighbourhoods)**.** *For a network $G=(V,E,\mathsf{w})$, as in Definition 3.2.1 and Definition 3.2.13, a set of* alternative neighbourhoods *is a collection of states:*

$$\Psi_\star = \{\Psi_\star(u) \subset \operatorname{span}\{|u,i\rangle : i \in L(u)\} : u \in V\}$$

such that for all $u \in V$,

$$|\psi_\star(u)\rangle := \frac{1}{\sqrt{\mathsf{w}_u}}\left(\sum_{i\in L^+(u)}\sqrt{\mathsf{w}_{u,i}}|u,i\rangle - \sum_{i\in L^-(u)}\sqrt{\mathsf{w}_{u,i}}|u,i\rangle\right) \in \Psi_\star(u).$$

We view the states of $\Psi_\star(u)$ as different possibilities for $|\psi_\star^G(u)\rangle$, only one of which is "correct." Let $d_{\max}=\max\{|L(u)| : u\in V\}$. We say we can generate *$\Psi_\star$ in complexity $\mathsf{A}_\star$, for some $\mathsf{A}_\star=\Omega(\log d_{\max})$, if there is a map $U_\star$ such that:*

- *for each $u\in V$, there is an orthonormal basis $\overline{\Psi}(u)=\{|\overline{\psi}_{u,0}\rangle,\dots,|\overline{\psi}_{u,a_u-1}\rangle\}$ for $\operatorname{span}\{\Psi_\star(u)\}$, such that for all $k\in[a_u]$, $U_\star|u,k\rangle=|\bar{\psi}_{u,k}\rangle$, and*
- *$U_\star$ can be implemented with complexity $\mathsf{A}_\star$.*

In Definition 4.2.1 we never exclude the possibility that the dimension a_u of the alternative neighbourhood $\Psi_\star(u)$ is equal to one, in which case we will assume without loss of generality that $\overline{\Psi}(u)=\{|\overline{\psi}_{u,0}\rangle\}=\{|\psi_\star(u)\rangle\}$. If that is the case, we will say that u has no additional alternative neighbourhoods. Without loss of generality, since $|\psi_\star(u)\rangle\in\Psi_\star(u)$, we assume that $|\psi_{u,0}\rangle=|\psi_\star(u)\rangle$ for any $\Psi_\star(u)$.

By adding the additional alternative neighbourhoods to the set $\Psi_{\mathcal{A}}$ spanning $\mathcal{A}$, we obtain

$$\Psi_{\mathcal{A}^{\mathsf{alt}}}=\{|\overline{\psi}_{u,i}\rangle : u\in V\setminus(\{s_0\}\cup M), i\in[a_u]\}.$$

We hence obtain the modified quantum walk operator

$$U_{\mathcal{A}^{\mathsf{alt}}\mathcal{B}} = (2\Pi_{\mathcal{A}^{\mathsf{alt}}}-I)(2\Pi_{\mathcal{B}}-I), \tag{4.1}$$

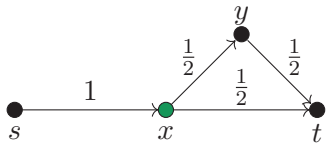
where $\Pi_{\mathcal{A}^{\mathsf{alt}}}$ is now the orthogonal projector onto $\mathcal{A}^{\mathsf{alt}}$ respectively, meaning

$$2\Pi_{\mathcal{A}^{\mathsf{alt}}}-I = 2\sum_{u\in V\setminus\{s_0,t\}}\sum_{i=0}^{a_u-1}|\overline{\psi}_{u,i}\rangle\langle\overline{\psi}_{u,i}|-I.$$

In our applications, these alternative neighbourhoods tackle the problem where it might be computationally easier to generate $\Psi_\star(u)$ instead of $|\psi_u\rangle$, which we elaborate on in Section 4.2.2 and 4.2.3. Essentially, these alternative neighbourhoods allow us to prove a version of Corollary 3.4.1 which we will state in Section 4.3, where we are able to (potentially drastically) reduce the cost of applying the walk operator $U_{\mathcal{AB}}$. However, we will see that this might come at the cost of increasing the effective resistance $\mathcal{R}_{s,t}$.

$\mathsf{w}_{u,v}$ for each $(u,v) \in \overrightarrow{E}$ $\theta_{u,v}$ for each $(u,v) \in \overrightarrow{E}$

Figure 4.1: Graph G with its s-t electrical flow θ.



$\theta^{\mathsf{alt}}_{u,v}$ for each $(u,v) \in \overrightarrow{E}$

Figure 4.2: Graph G where the coloured vertex x has additional alternative neighbourhoods (see (4.2)). A unit s-t flow θ^{alt} is displayed, whose corresponding flow state forms a 0-positive witness with respect to these extra alternative neighbourhoods

4.2.1 Example graph

To see how alternative neighbourhoods work, we once again return to our network from Figure 3.1 (without edge labels). Recall that each edge $(u,v) \in \overrightarrow{E}$ has a weight of $\mathsf{w}_{u,v} = 1/4$, except for the edge (s,x), which has a weight of $\mathsf{w}_{s,x} = 1$. This is visualised here in Figure 4.1, along with the s-t electrical flow θ on G.

Now suppose that for some abstract reason it is computationally infeasible to generate the star state $|\psi_\star(x)\rangle$, but it is straightforward to generate the following set of alternative neighbourhoods for x:

$$\begin{aligned}\Psi_\star(x) &= \left\{|\psi_\star(x)\rangle, |\psi^{\mathsf{alt}}_\star(x)\rangle\right\} \\ &= \left\{\sqrt{\frac{2}{3}}\left(-|x,s\rangle + \frac{1}{2}|x,y\rangle + \frac{1}{2}|x,t\rangle\right), \sqrt{\frac{2}{3}}\left(\frac{1}{2}|x,s\rangle - |x,y\rangle + \frac{1}{2}|x,t\rangle\right)\right\}.\end{aligned} \tag{4.2}$$

We could have added the third option $\sqrt{\frac{2}{3}}\left(-|x,s\rangle + \frac{1}{2}|x,y\rangle + \frac{1}{2}|x,t\rangle\right)$ to $\Psi_\star(x)$ as well, but this state is a linear combination of $|\psi_\star(x)\rangle$ and $|\psi^{\mathsf{alt}}_\star(x)\rangle$ and is hence already contained in $\mathcal{A}^{\mathsf{alt}}$. If we were to now try and apply Theorem 3.3.8, we could no longer use the flow state $|\theta\rangle$ of the s-t electrical flow θ on G as a 0-positive witness: in (3.31), we saw how $\langle\psi_\star(x)|\theta\rangle = 0$, which is required for $|\theta\rangle$ to lie in $\mathcal{A}^\perp$ by Definition 3.3.5. However, the same 0-positive witness would not work for $\mathcal{A}^{\mathsf{alt}}$, since

$$\langle\psi^{\mathsf{alt}}_\star(x)|\theta\rangle = \frac{1}{\sqrt{2\mathcal{E}(\theta)}}\left(\sqrt{\frac{2}{3}}\frac{1}{2}\langle x,s|x,s\rangle - \sqrt{\frac{2}{3}}\frac{2}{3}\langle x,y|x,y\rangle + \sqrt{\frac{2}{3}}\frac{1}{2}\frac{4}{3}\langle x,t|x,t\rangle\right) = \frac{1}{2\sqrt{11}}.$$

There does exist another flow θ^{alt} however, displayed in Figure 4.2, which could potentially work as 0-positive witness as it satisfies $\langle\psi^{\mathsf{alt}}_\star(x)|\theta^{\mathsf{alt}}\rangle = 0$. In Section 6.2.4 we show that it actually corresponds to the optimal 0-positive witness with respect to $\Pi_{\mathcal{A}^{\mathsf{alt}}}$. Intuitively, this means that it is not enough to have any "regular" s-t flow θ, which by Kirchhoff's Law is conserved at all the intermediate vertices (see Definition 3.2.5). Instead, we need a flow θ^{alt} which is conserved with respect to all alternative neighbourhoods.

Figure 4.3: Graph G where the coloured vertex x has additional alternative neighbourhoods (see (4.2)). The unique unit s-t flow θ is displayed, but its corresponding flow state does not form a 0-positive witness with respect to these extra alternative neighbourhoods.

Now depending on the alternative neighbourhoods in $\Psi_\star$, there may not exist any 0-positive witness at all to apply Theorem 3.3.8. To exhibit such a counterexample, we modify G once more, this time removing the edge (y,t) from $\overrightarrow{E}$, which is visualised in Figure 4.3. It is clear that any unit s-t flow θ must satisfy $\theta_{s,x} = \theta_{x,t} = 1$ and $\theta_{x,y} = 0$, but in doing so we find that $|\theta\rangle$ is not orthogonal to $|\psi_\star^{\mathsf{alt}}(x)\rangle$:

$$\langle \psi_\star^{\mathsf{alt}}(x)|\theta\rangle = \frac{1}{\sqrt{2\mathcal{E}(\theta^{\mathsf{alt}})}} \left(\sqrt{\frac{2}{3}}\frac{1}{2}\langle x,s|x,s\rangle + \sqrt{\frac{2}{3}}\frac{1}{2}\langle x,t|x,t\rangle \right) = \frac{1}{3}.$$

4.2.2 Welded trees

We motivate the alternative neighbourhoods modification by an application to the *welded trees* problem [CCD+03]. In the welded trees problem, the input is an oracle O_G for a graph G with $s,t \in V \subset \{0,1\}^{2n}$. Each of s and t is the root of a full binary tree with 2^n leaves, and we connect these leaves with a pair of random matchings. This results in a graph in which all vertices except s and t have degree 3, and s and t each have degree 2. Given a string $u \in \{0,1\}^{2n}$, the oracle O_G returns $\perp$ if $u \notin V$, which is true for all but at most a 2^{-n+2} fraction of strings, and otherwise it returns a list of the 2 or 3 neighbours of u. We assume $s = 0^{2n}$, so we can use s as our starting point, and the goal is to find t, which we can recognise since it is the only other vertex with only 2 neighbours. The classical query complexity of this problem is $2^{\Omega(n)}$ [CCD+03]. Intuitively, that is because this problem is set up so that a classical algorithm has no option but to do a random walk, starting from s, until it hits t. However, this takes $2^{\Omega(n)}$ steps, because wherever a walker is in the graph, the probability of moving towards the centre, where the leaves of the two trees are connected, is twice the probability of moving away from the centre, towards s or t. So a walker quickly moves from s to the centre, but then it takes exponential time to escape to t.

In [CCD+03] a continous quantum algorithm was shown that solves this problem in $\mathrm{poly}(n)$, later improved to $O(n^{1.5})$ by [AC21]. However if we try to reproduce this result in the electric network framework, we will get an exponential-time algorithm, essentially because the total weight of the graph is exponential if we set every edge weight to 1.

Suppose we could add weights to the edges of G, so that at any vertex u, the probability of moving towards the centre or away from the centre were the same: that is, if w is the weight on the edge from u to its *parent*, then the other two edges should have weight $\mathsf{w}/2$. This would already be very helpful for a classical random walk, however, a bit of thought shows that this is not possible to implement. By querying u, we learn the labels of its three neighbours, v_1, v_2, v_3, which are random $2n$-bit strings, but we get no indication which is the parent. However, we know that the correct star state in the weighted graph that we

would like to be able to walk on is proportional to one of the following:

$$|u,v_1\rangle + \frac{1}{2}|u,v_2\rangle + \frac{1}{2}|u,v_3\rangle, \quad |u,v_2\rangle + \frac{1}{2}|u,v_1\rangle + \frac{1}{2}|u,v_3\rangle, \quad |u,v_3\rangle + \frac{1}{2}|u,v_1\rangle + \frac{1}{2}|u,v_2\rangle.$$

Thus, we add all three states (up to some minor modifications) as alternative neighbourhoods to $\Psi_\star(u)$, which yields an algorithm that can learn any bit of information about t in $O(n)$ queries. By composing this with the Bernstein-Vazirani algorithm [BV97] we can find t. For details, see Section 4.4.

We emphasise that our application to the welded trees problem does not use the edge composition technique. It would be trivial to embed any known exponential speedup in our framework by simply embedding the exponentially faster quantum algorithm in one of the edges of the graph, but we are able to solve the welded trees problem using only the alternative neighbourhoods idea.

4.2.3 3-Distinctness

We describe an attempt at a quantum walk algorithm for 3-distinctness, how it fails, and how the multidimensional quantum walk framework comes to the rescue. While our result for $k = 3$ is not new, our generalisation to $k > 3$ is, and the case of $k = 3$ is already sufficient to illustrate our techniques. Formally, the problem of 3-distinctness is: given a string $x \in [q]^n$, output a 1 if and only if there exist distinct $a_1, a_2, a_3 \in [n]$ such that $x_{a_1} = x_{a_2} = x_{a_3}$. We make the standard simplifying assumptions (without loss of generality) that if such a 3-collision exists, it is unique, and moreover, there is an equipartition $[n] = A_1 \cup A_2 \cup A_3$ such that $a_1 \in A_1$, $a_2 \in A_2$ and $a_3 \in A_3$.

We now describe a graph that will be the basis for a quantum walk attempt. A vertex v_{R_1,R_2} is described by a pair of sets $R_1 \subset A_1$ and $R_2 \subset A_2$. v_{R_1,R_2} stores these sets, as well as input-dependent *data* consisting of the following:

- Queried values for all of R_1: $D_1(R) := \{(i, x_i) : i \in R_1\}$.
- Queried values for those elements of R_2 that have a match in R_1:
$$D_2(R) := \{(i_1, i_2, x_{i_1}) : i_1 \in R_1, i_2 \in R_2, x_{i_1} = x_{i_2}\}.$$

By only keeping track of the values in R_2 that have a match in R_1, we save the cost of initially querying the full set R_2. The vertices will be in 4 different classes, for some parameters r_1 and r_2 with $r_1 \ll r_2$:

$$\begin{aligned}
V_0 &= \{v_{R_1,R_2} : |R_1| = r_1, |R_2| = r_2\} \\
V_1 &= \{v_{R_1,R_2} : |R_1| = r_1 + 1, |R_2| = r_2\} \\
V_2 &= \{v_{R_1,R_2} : |R_1| = r_1 + 1, |R_2| = r_2 + 1\} \\
V_3 &= \{v_{R_1,R_2,i_3} : |R_1| = r_1 + 1, |V_2| = r_2 + 1, i_3 \in A_3\}.
\end{aligned}$$

The vertices $v_{R_1,R_2,i_3} \in V_3$ are just like the vertices in V_2, except there is an additional index $i_3 \in A_3$ stored. We connect vertices in V_ℓ and $V_{\ell+1}$ in the obvious way: $v_{R_1,R_2} \in V_\ell$ is adjacent to $v_{R_1',R_2'} \in V_{\ell+1}$ if and only if $R_1 \subseteq R_1'$ and $R_2 \subseteq R_2'$ (exactly one of these inclusions is proper); and $v_{R_1,R_2} \in V_2$ is adjacent to $v_{R_1,R_2,i_3} \in V_3$ for any $i_3 \in A_3$ (see Figure 4.4). We say a vertex $v_{R_1,R_2,i_3} \in V_3$ is marked if $a_1 \in R_1$, $a_2 \in R_2$, and $a_3 = i_3$, where (a_1, a_2, a_3) is the unique 3-collision. Thus, a quantum walk that decides if there is a marked vertex or not decides 3-distinctness.

We imagine a quantum walk that starts in a uniform superposition over V_0. To construct this initial state, we first take a uniform superposition over all sets R_1 of r_1 indices, and query them. Next we take a uniform superposition over all sets R_2 of size r_2, but rather than query everything in R_2, we search for all indices in R_2 that have a match in

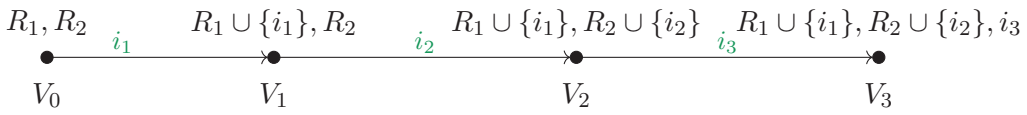


Figure 4.4: A sample path from V_0 to V_3 in our first attempt at a quantum walk for 3-distinctness. The coloured indices can be seen to label the edges.

R_1. This saves us the cost of querying all r_2 elements of R_2, which is important because we will set r_2 to be larger than the total complexity we aim for (in this case, $r_2 \gg n^{5/7}$), so we could not afford to spend so much time. However, we do not only care about query complexity, but also the total time spent on non-query operations, so we also do not want to spend time writing down the set R_2, even if we do not query it, which is the first problem with this approach:

Problem 1: Writing down R_2 would take too long.

The fix for Problem 1 is rather simple: we will not let R_2 be a uniform random set of size r_2. Instead, we will assume that A_2 is partitioned into m_2 blocks, each of size $n/(3m_2)$, and R_2 will be made up of $t_2 := 3m_2r_2/n$ of these blocks. This also means that when we move from V_1 to V_2, we will add an entire block, rather than just a single index. The main implication of this is that when we move from V_1 to V_2, we will have to search the new block of indices that we are adding to R_2 for any index that collides with R_1. This means that transitions from V_1 to V_2 have a non-trivial cost, n^ε for some small constant ε, unlike all other transitions, which have polylogarithmic cost. Naively we would incur a multiplicative factor of n^ε on the whole algorithm, but we avoid this because the edge composition technique essentially allows us to only incur the cost n^ε on the edges that actually incur this cost, and not on every edge in the graph. Otherwise, our solution to Problem 1 is technical, but not deep, and so we gloss over Problem 1 and its solution for the remainder of this high-level synopsis. This is the only place we use the edge composition part of the framework in our applications, but we suspect it can be used in much more interesting ways.

Moving on, in order to take a step from a vertex $v_{R_1,R_2} \in V_0$ to a vertex $v_{R_1\cup\{i_1\},R_2} \in V_1$, we need to select a uniform new index i_1 to add to R_1, and then also update the data we store with each vertex. That means we have to query i_1 and add (i_1, x_{i_1}) to $D_1(R)$, which is simple, and can be done in $O(\log n)$ basic operations as long as we use a reasonable data structure to store $D_1(R)$; and we also have to update $D_2(R)$ by finding anything in R_2 that collides with i_1. Since R_2 has not been queried, this latter update would require an expensive search, which we do not have time for, so we want to avoid this. However, if we do not search R_2 for any i_2 such that $x_{i_2} = x_{i_1}$, then whenever we add some i_1 that has a match in R_2, the data becomes incorrect, and we have introduced what is referred to in [Bel12a] as a *fault*. This is a serious issue, because if i_1 is the unique index in R_1 such that there exists $i_2 \in R_2$ with $x_{i_1} = x_{i_2}$, but this is not recorded in $D_2(R)$, then i_1 is "remembered" as having been added after i_2. That is, the resulting vertex does not only depend on $R_1 \cup \{i_1\}, R_2$, but on i_1 as well. For quantum interference to happen, it is crucial that when we are at a vertex v, the state does not remember anything about how we got there.

Problem 2: When we add i_1 to R_1 without searching for a match in R_2, we may introduce a *fault*.

Our handling of this is inspired by the solution to an analogous problem in the query upper bound of [Bel12a]. We partition R_1 into three sets: $R_1(\{1\})$, $R_1(\{2\})$, and $R_1(\{1,2\})$; and

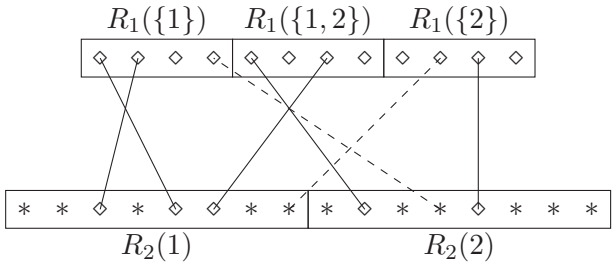


Figure 4.5: The data we keep track of for a vertex v_{R_1,R_2}. $\diamond$ represents a queried index. $*$ represents an index whose query value is not stored. We only store the query value of an index in $R_2(s)$ if it collides with something in $R_1(\{s\}) \cup R_1(\{1,2\})$, shown here by a solid line. If $i_2 \in R_2(1)$ collides with some value in $R_1(\{2\})$, shown here by a dashed line, we do not record that, and do not store x_{i_2}.

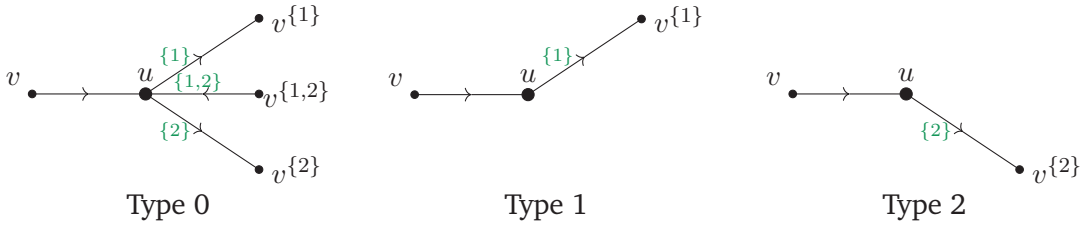


Figure 4.6: The possible neighbourhoods of $u = v_{R_1,R_2,i_1} \in V_0^+$, depending on the type of vertex. $v^S \in V_1$ is obtained from v by adding i_1 to $R_1(S)$. The backwards neighbour $v = v_{R_1,R_2} \in V_0$ is always the same.

R_2 into two sets $R_1(1)$ and $R_1(2)$. Then $D_2(R)$ will only store collisions (i_1, i_2, x_{i_1}) such that $x_{i_1} = x_{i_2}$ if $i_1 \in R_1(S)$ and $i_2 \in R_2(s)$ for some $s \in S$. This is shown in Figure 4.5.

Now when we add i_1 to R_1, we have three choices: we can add it to $R_1(\{1\})$, $R_1(\{2\})$, or $R_1(\{1,2\})$. Importantly, at least one of these choices does not introduce a fault. To see this, suppose there is some $i_2 \in R_2$ such that $x_{i_1} = x_{i_2}$. We claim there can be at most one such index, because otherwise there would be a 3-collision in $A_1 \cup A_2$, and we are assuming the unique 3-collision has one part in A_3. This leads to three possibilities:

Type 1: $i_2 \in R_2(2)$, in which case, adding i_1 to $R_1(\{1\})$ does not introduce a fault.

Type 2: $i_2 \in R_2(1)$, in which case, adding i_1 to $R_1(\{2\})$ does not introduce a fault.

Type 0: There is no such i_2, in which case, adding i_1 to $R_1(\{1\})$ or $R_1(\{2\})$ or $R_1(\{1,2\})$ does not introduce a fault.

We modify the graph so that we first move from $v_{R_1,R_2} \in V_0$ to $v_{R_1,R_2,i_1} \in V_0^+$ by selecting a new $i_1 \in A_1 \setminus R_1$, and then move from v_{R_1,R_2,i_1} to $v_{R_1 \cup \{i_1\},R_2} \in V_1$ – here there are three possibilities for $R_1 \cup \{i_1\}$, depending on to which of the three parts of R_1 we add i_1. However, we will only add i_1 to a part of R_1 that does not introduce a fault. Thus, a vertex v_{R_1,R_2,i_1} in V_0^+ has one edge leading back to V_0, and either one or three edges leading forward to V_1, as shown in Figure 4.6.

On its own, this is not a solution, because for a given v_{R_1,R_2,i_2}, in order to determine its type, we would have to search for an $i_2 \in R_2$ such that $x_{i_1} = x_{i_2}$, which is precisely what we want to avoid. However, this is exactly the situation where the alternative neighbourhood technique is useful. For all $u \in V_0^+$, we will let $\Psi_\star(u)$ contain all three possibilities shown in Figure 4.6, of which exactly one is the correct state. We are then able to carefully construct a flow that is orthogonal to all three states, in our analysis. The idea is that all

incoming flow from v must leave along the edge $(u, v^{\{1\}})$ so that the result is a valid flow in case of Type 1. However, in order to be a valid flow in case of Type 2, all incoming flow from v must leave along the edge $(u, v^{\{2\}})$. But now to ensure that we also have a valid flow in case of Type 0, we must have negative flow on the edge $(u, v^{\{1,2\}})$, or equivalently, flow from $v^{\{1,2\}}$ to u. This is indicated by the arrows on the edges in Figure 4.6. For details, see Section 5.4.

4.3 The multidimensional quantum walk framework

In this section, we present the multidimensional quantum walk framework, by constructing a phase estimation algorithm as in Definition 3.3.1, whose parameters incorporate both the techniques of edge composition and alternative neighbourhoods. As discussed in Section 3.4, we will use a slightly different initialisation of the phase estimation parameters, since our framework will deal with general edge labels and will have to construct networks with length (see Definition 3.2.7) to incorporate the edge composition technique. We summarise this initialisation in Table 4.1.

4.3.1 The transition subroutine

To incorporate the edge composition technique, first recall from Definition 3.5.1 that a quantum subroutine is given by a sequence $U_0, \dots, U_{\mathsf{T}_{\max}-1}$ of unitaries on some abstract Hilbert space $\hat{\mathcal{H}} = \operatorname{span}\{|z\rangle : z \in \mathcal{Z}\}$, such that we can implement $\sum_{t=0}^{\mathsf{T}_{\max}-1} |t\rangle\langle t| \otimes U_t$ in cost $\operatorname{polylog}(\mathsf{T}_{\max})$. In our case, the subroutine computes the *transition map* (see Definition 3.2.13), $|u, i\rangle \mapsto |v, j\rangle$, so we assume

$$\{(u, i) : u \in V, i \in L(u)\} \subseteq \mathcal{Z}.$$

Moreover, we allow these subroutines to run with some bounded error, meaning for any $(u, v) \in \overrightarrow{E}$, with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$, we have

$$\left\| |v, j\rangle - U_{\mathsf{T}_{u,v}-1} \dots U_0 |u, i\rangle \right\|^2 = \epsilon_{u,v}. \tag{4.3}$$

We also assume that we can bound this error by some ϵ, except for some small subset of the edges denoted by $\tilde{E}$. So $\epsilon_{u,v} \leq \epsilon$ whenever $(u, v) \in \overrightarrow{E} \setminus \tilde{E}$. Otherwise, we only have the trivial upper bound $\epsilon_{u,v} \leq 4$.

Furthermore, we will assume that in $O(1)$ time, we can check, for any $z \in \mathcal{Z}$, if $z = (u, i)$ for some $u \in V$ and $i \in L(u)$, and further, whether $i \in L^+(u)$ or $i \in L^-(u)$. This is without loss of generality, by the following construction. Assume that for all $u \in V$, every label in $L^+(u)$ ends with the symbol $\rightarrow$, and every label in $L^-(u)$ ends with the symbol $\leftarrow$. Further assume that no other $z \in \mathcal{Z}$ ends with these symbols. Then it is sufficient to check a single constant-dimensional register.

Lastly, we will assume that $\mathsf{T}_{u,v}$ is always even. This assumption incurs at most a small constant slowdown, but we shall see that it guarantees that the network (with length) on which the quantum walk takes place will always be bipartite. Without loss of generality, we can assume that after exactly $\mathsf{T}_{u,v}$ steps, the algorithm sets an *internal flag register* to 1, and we will let this 1-flag be part of the final state (v, j) by letting each $i \in L(u)$ contain an extra bit set to 1. This also ensures that the state of the algorithm is never $|v, j\rangle$ before $\mathsf{T}_{u,v}$ steps have passed. The reason why this is without loss of generality, is because we can simply let the algorithm use an internal timer in order to decide to set a flag after exactly $\mathsf{T}_{u,v}$ steps, and uncompute this timer using our ability to compute $\mathsf{T}_{u,v}$ from the final correct state $|v, j\rangle$.

Recall from Definition 3.5.1 that for any $u \in V$, $i \in L(u)$ and $t \in [\mathsf{T}_{\max} - 1]_0$,

$$U_t \dots U_0 |u, i\rangle \in \operatorname{span}\{|z\rangle : z \in \mathcal{Z}_{u,i}\}.$$

For convenience, we will let $\mathcal{Z}_{u,v} = \mathcal{Z}_{u,i}$, where $v = f_u(i)$. For $b \in \{0, 1\}$, let $\mathcal{Z}^b_{u,v} \subset \mathcal{Z}_{u,v}$ be the subset of states in which the algorithm's internal flag register is set to b. So by the above discussion, we have $(v, j) \in \mathcal{Z}^1_{u,v}$,

$$\forall t \in [\mathsf{T}_{u,v} - 1]_0,\ U_t \dots U_0 |u, i\rangle \in \mathcal{Z}^0_{u,v}, \text{ and } \forall t \geq \mathsf{T}_{u,v},\ U_t \dots U_0 |u, i\rangle \in \mathcal{Z}^1_{u,v}.$$

4.3.2 Parameters of the phase estimation algorithm

Hilbert space: Our phase estimation algorithm will again work on a Hilbert space related to the modified graph G' as shown Figure 3.2, where the extra vertex s_0 is added to our graph G via the directed edges (s_0, u) of weight $\mathsf{w}_0\sigma(u)$ for $u \in V_0 \subset V(G)$. For the more general case, basis states of the form $|u, i\rangle$ containing the edge labels must now also be incorporated into the Hilbert space, as well as the subroutine states of Section 4.3.1. So instead of the space defined in (3.22), our algorithm will work on the space:

$$\begin{aligned}\mathcal{H} = \operatorname{span}\{|u, i\rangle|0\rangle : u \in V(G), i \in L^+(u) \cup \{0\}\} \oplus \operatorname{span}\{|v, j\rangle|0\rangle : v \in V(G), j \in L^-(v)\} \\ \oplus \bigoplus_{(u,v) \in \overrightarrow{E}(G)} \operatorname{span}\{|z\rangle|t\rangle : z \in \mathcal{Z}^0_{u,v}, t \in [\mathsf{T}_{u,v} - 1]\} \cup \{|z\rangle|\mathsf{T}_{u,v}\rangle : z \in \mathcal{Z}^1_{u,v}\}.\end{aligned} \tag{4.4}$$

The first span in (4.4) will contain the outgoing edges, the second the incoming edges and the third span is to deal with the transition subroutines. One can see that we assume that for all $u \in V_0$, the edge to s_0 is labelled by $0 \notin L(u)$.

Observe that since the edge labels can be distinct from the vertices themselves, the existence of a $|u, i\rangle \in \mathcal{H}$ does not immediately imply the existence of a state $|i, u\rangle$ in $\mathcal{H}$. More specifically, this means that we will not be reflecting around the antisymmetric subspace of $\mathcal{H}$ like we did in Section 3.4. Instead, the basis states corresponding to the transition subroutines will aide in constructing the second reflection, as we will see shortly.

Star states: The construction of our star states will have to be slightly modified compared to (3.23), due to the different choice of Hilbert space. Suppose we have a set of alternative neighbourhoods $\Psi_\star$ for the network G as in Definition 4.2.1. To make sure that $\Psi_{\mathcal{A}}$ and $\Psi_{\mathcal{B}}$ are subsets of $\mathcal{H}$, we append a register $|0\rangle$ to all states in $\Psi_\star$. Note that even though $\Psi_{\mathcal{A}}$ (and even $\Psi_{\mathcal{B}}$) might contain additional alternative neighbourhoods, we do note use the notation $\mathcal{A}^{\mathsf{alt}}$ in the subscript, as there is no ambiguity with some other set $\Psi_{\mathcal{A}}$.

Second of all, we must not forget that our phase estimation algorithm takes place on the Hilbert space associated to the modified graph G', meaning the star state of any vertex $u \in V_0$, must have an incoming edge from s_0. So for any $u \in V_0$, the star state (see Definition 3.2.13 or Definition 4.2.1) of u with respect to G' is:

$$\sqrt{\mathsf{w}^{G'}_u}|\psi^{G'}_\star(u)\rangle|0\rangle = \underbrace{\sum_{i \in L^+(u)} \sqrt{\mathsf{w}_{u,i}}|u, i\rangle|0\rangle - \sum_{i \in L^-(u)} \sqrt{\mathsf{w}_{u,i}}|u, i\rangle|0\rangle}_{= \sqrt{\mathsf{w}^G_u}|\psi^G_\star(u)\rangle|0\rangle} - \sqrt{\mathsf{w}_0\sigma(u)}|u, 0\rangle|0\rangle. \tag{4.5}$$

We write $\Psi'_\star$ for the set of alternative neighbourhoods $\Psi_\star$ after it has been "lifted" to $\mathcal{H}$, meaning if we assume that any vertex $u \in V_0$ does not contain any additional alternative

neighbourhoods, we have

$$\Psi'_\star := \bigcup_{u\in V(G)\setminus(V_0\cup M)} \underbrace{\{|\psi\rangle|0\rangle : |\psi\rangle \in \Psi_\star(u)\}}_{=:\Psi'_\star(u)} \cup \bigcup_{u\in V_0} \underbrace{\left\{|\psi_\star^{G'}(u)\rangle|0\rangle\right\}}_{=:\Psi'_\star(u)}. \tag{4.6}$$

Transition states: For each $u \in V(G)$ and $i \in L^+(u)$, define a state

$$|\psi^{u,i}_{\rightarrow}\rangle := |u,i\rangle|0\rangle - U_0|u,i\rangle|1\rangle. \tag{4.7}$$

These represent a transition from an outgoing edge to the first step of the algorithm implementing that edge transition. For each $(u,v) \in \overrightarrow{E}(G)$, and $t \in [\mathsf{T}_{u,v} - 1]$, define states:

$$\Psi_t^{u,v} := \left\{|\psi_t^z\rangle := |z\rangle|t\rangle - U_t|z\rangle|t+1\rangle : z \in \mathcal{Z}^0_{u,v}\right\}. \tag{4.8}$$

These represent steps of the edge transition subroutine. For each $v \in V(G)$ and $j \in L^-(v)$, with $u = f_v(j)$, define a state:

$$|\psi^{v,j}_{\leftarrow}\rangle := |v,j\rangle|\mathsf{T}_{u,v}\rangle - |v,j\rangle|0\rangle. \tag{4.9}$$

These represent exiting the algorithm to an edge going into vertex v. Letting $\Psi'_\star$ be as in (4.6), define

$$\begin{aligned}
\Psi_{\mathcal{A}} &= \Psi'_\star \cup \bigcup_{(u,v)\in\overrightarrow{E}(G)} \bigcup_{\substack{t=1:\\ t \text{ odd}}}^{\mathsf{T}_{u,v}-1} \Psi_t^{u,v} \\
\Psi_{\mathcal{B}} &= \bigcup_{u\in V(G)} \{|\psi^{u,i}_{\rightarrow}\rangle : i \in L^+(u)\} \cup \{|\psi^{u,j}_{\leftarrow}\rangle : j \in L^-(v)\} \cup \bigcup_{(u,v)\in\overrightarrow{E}(G)} \bigcup_{\substack{t=1:\\ t \text{ even}}}^{\mathsf{T}_{u,v}-1} \Psi_t^{u,v}.
\end{aligned} \tag{4.10}$$

The reason we have divided the states in this way between $\Psi_{\mathcal{A}}$ and $\Psi_{\mathcal{B}}$ is so that if we replace each $\Psi_\star(u)$ with an orthonormal basis, all states in $\Psi_{\mathcal{A}}$ (or $\Psi_{\mathcal{B}}$) are pairwise orthogonal. We leave it up to the reader to verify that this is the case (it is implicitly proven in Section 4.3.3), but we note that this fact relies on the assumption that $\mathsf{T}_{u,v}$ is always even. This ensures that for even t, $\langle t+1|\mathsf{T}_{u,v}\rangle = 0$, so $\langle\psi_t^z|\psi^{v,j}_{\leftarrow}\rangle = 0$. Figure 4.7 shows a graph of the overlap between various sets of states, and we can observe that the sets in $\Psi_{\mathcal{A}}$ and the sets in $\Psi_{\mathcal{B}}$ form a bipartition of this overlap graph into independent sets.

In Section 4.4 we will compare this with the initialisation from Section 3.4, which intuitively could be seen as the case where each $\mathsf{T}_{u,v} = 0$. If not for the labels, nor the extra register containing the internal timer, reflecting around the state in $\Psi_{\mathcal{B}}$ can be understood as reflecting around the antisymmetric subspace. However, this interpretation is only meant to provide intuition, as the transition subroutine definitions become ill-defined when $\mathsf{T}_{u,v} = 0$.

Initial state: For the initial state of the algorithm, in (3.28) we chose the normalised projection of $|\psi_\star(s_0)\rangle$ onto $\mathcal{B}^\perp$. Since both our star states, as well as $\mathcal{B}^\perp$ have changed compared to Section 3.4, we have to rederive this definition. First of all, as we did with the star state of each $u \in V(G)$, we append an extra register to ensure that it is an element of $\mathcal{H}$:

$$|\psi_\star(s_0)\rangle|0\rangle = \sum_{u\in V_0} \sqrt{\sigma(u)}|s_0,u\rangle|0\rangle.$$

The problem is however that $|s_0,u\rangle$ is not a state in $\mathcal{H}$, but $|u,0\rangle$ is:

$$\sum_{u\in V_0} \sqrt{\sigma(u)}|u,0\rangle|0\rangle.$$

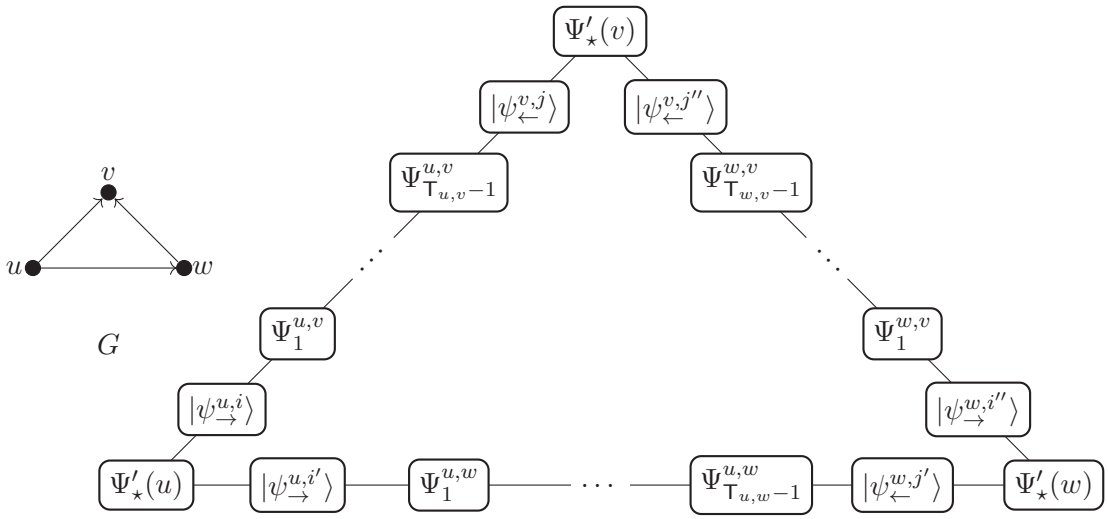


Figure 4.7: A graph showing the overlap of various sets of states, for an example graph G. With the exception of the spaces $\Psi'_\star(u)$ (which we will replace with orthonormal bases in Section 4.3.3), each node represents an orthonormal set. There is an edge between two nodes if and only if the sets contain overlapping vectors.

For the projection onto $\mathcal{B}^\perp$, note that the each edge coming out of s_0 is not an element of $E(G)$. It is therefore straightforward to see that each basis state $|u,0\rangle$ already does not overlap with any of states spanning $\Psi_\mathcal{B}$, meaning we can set our initial state as

$$|\psi_0\rangle = \sum_{u\in V_0} \sqrt{\sigma(u)}|u,0\rangle|0\rangle. \tag{4.11}$$

The flow state: Although the flow state is not a parameter of our phase estimation algorithm, but instead an ingredient to construct the positive witness as discussed in Section 3.4.2. However, due to how we have changed the parameters in this section so far, we also discuss how this affects our flow state compared to its construction in (3.29):

$$\frac{1}{\sqrt{2\mathcal{E}(\theta)}} \sum_{(u,v)\in\vec{E}(G)} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}} \left(|u,v\rangle + |v,u\rangle\right).$$

This state is currently not a state in $\mathcal{H}$ due to the lack of labels in the second register. This can be resolved by considering the state

$$\frac{1}{\sqrt{2\mathcal{E}(\theta)}} \sum_{(u,v)\in\vec{E}(G)} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}} \left(|u,f_u^{-1}(v)\rangle + |v,f_u^{-1}(u)\rangle\right).$$

Due to our different initialisation of the flow state and the subspace $\mathcal{B}$, it might be that this new flow state construction now longer lives in $\mathcal{B}^\perp$. Although it is not immediately clear yet (we will prove this in Corollary 4.3.10 as part of the prove of our general framework), this state is indeed orthogonal (if the transition subroutines run without error, otherwise it is "almost" orthogonal) to $\mathcal{B}$. Hence, under our parameters as defined in (4.10), we define the flow state of any unit s-t flow θ on G as

$$|\theta\rangle := \frac{1}{\sqrt{2\mathcal{E}(\theta)}} \sum_{(u,v)\in\vec{E}(G)} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}} \left(|u,f_u^{-1}(v)\rangle + |v,f_u^{-1}(u)\rangle\right). \tag{4.12}$$

4.3.3 The framework

We now state our general multidimensional quantum walk framework.

Theorem 4.3.1 (Multidimensional Quantum Walk Framework)**.** *Fix a family of networks G that may depend on some implicit input x, with disjoint sets $V_0, V_{\mathsf{M}} \subset V(G)$ such that for any vertex, checking if $v \in V_0$ (resp. if $v \in V_{\mathsf{M}}$) can be done in at most $\mathsf{A}_\star$ complexity. Let $M \subseteq V_{\mathsf{M}}$ be the* marked set*, and σ an* initial distribution *on V_0. Let $\Psi_\star = \{\Psi_\star(u) : u \in V(G)\}$ be a set of alternative neighbourhoods for G (see Definition 4.2.1). For all $u \in (V_0 \cup V_{\mathsf{M}})$, assume that $\Psi_\star(u) = \{|\psi_\star^G(u)\rangle\}$. Fix some positive real-valued $\mathcal{W}^{\mathsf{T}}$ and $\mathcal{R}^{\mathsf{T}}$, that may scale with $|x|$. Suppose the following conditions hold.*

Setup Subroutine: *The state $|\sigma\rangle = \sum_{u\in V_0} \sqrt{\sigma(u)}|u\rangle$ can be generated in cost S, and furthermore, for any $u \in V_0$, $\sigma(u)$ can be computed in $O(1)$ complexity.*

Star State Generation Subroutine: *We can generate $\Psi_\star$ in complexity $\mathsf{A}_\star$.*

Transition Subroutine: *There is a quantum subroutine (see Definition 3.5.1) that implements the transition map of G (see Definition 3.2.13) with errors $\{\epsilon_{u,v}\}_{(u,v)\in\overrightarrow{E}(G)}$ and costs $\{\mathsf{T}_{u,v}\}_{(u,v)\in\overrightarrow{E}(G)}$. We make the following assumptions on the errors $\epsilon_{u,v}$, where $\tilde{E} \subset \overrightarrow{E}(G)$ is some (possibly unknown) set of edges on which we allow the subroutine to fail:*

- **TS1** *For all $(u,v) \in \overrightarrow{E}(G) \setminus \tilde{E}$, $\epsilon_{u,v} \leq \epsilon$, where $\epsilon = o\left(\frac{1}{\mathcal{W}^{\mathsf{T}}\mathcal{R}^{\mathsf{T}}}\right)$.*
- **TS2** *For all $(u,v) \in \tilde{E}$, there is no non-trivial upper bound on $\epsilon_{u,v}$, but we have that $\widetilde{\mathcal{W}} := \sum_{e\in\tilde{E}} \mathsf{w}_e = o\left(\frac{1}{\mathcal{R}^{\mathsf{T}}}\right)$.*

Checking Subroutine: *There is an algorithm that checks, for any $u \in V_{\mathsf{M}}$, if $u \in M$, in cost $\mathsf{A}_\star$.*

Positive Condition: *Interpreting $\mathsf{T}_{u,v}$ as a length function on $\overrightarrow{E}(G)$, G^{T} is the graph obtained by replacing each edge (u,v) of G with a path of length $\mathsf{T}_{u,v}$ (see Definition 3.2.7 and Figure 4.7). If $M \neq \emptyset$, then there exists a unit flow θ on G (see Definition 3.2.3) with flow state $|\theta\rangle$ (see (4.12)) such that*

- **P1** *For all $e \in \tilde{E}$, $\theta_e = 0$.*
- **P2** *For all $u \in V(G) \setminus (V_0 \cup M)$ and $|\psi_\star\rangle \in \Psi_\star(u)$, $\langle\psi_\star|\theta\rangle = 0$.*
- **P3** $\sum_{u\in V_0} \theta_u = 1$.
- **P4** $\sum_{u\in V_0} \frac{|\theta_u - \sigma(u)|^2}{\sigma(u)} \leq 1$.
- **P5** $\mathcal{E}^{\mathsf{T}}(\theta) \leq \mathcal{R}^{\mathsf{T}}$.

Negative Condition: *If $M = \emptyset$, then $\mathcal{W}(G^{\mathsf{T}}) \leq \mathcal{W}^{\mathsf{T}}$.*

Then there is a quantum algorithm that decides if $M = \emptyset$ or not with bounded error in complexity:

$$O\left(\mathsf{S} + \sqrt{\mathcal{R}^{\mathsf{T}}\mathcal{W}^{\mathsf{T}}}\left(\mathsf{A}_\star + \mathsf{polylog}(\mathsf{T}_{\max})\right)\right).$$

In the remainder of this section, we prove Theorem 4.3.1 in a similar fashion as we proved Corollary 3.4.1 in Section 3.4.2. The difference is that the parameters of our phase estimation algorithm will slightly differ, as we need to deal with the alternative neighbourhoods and transition subroutines. We first provide some additional intuition to the conditions stated in Theorem 4.3.1.

The **Setup Subroutine** and **Star State Generation Subroutine** are straightforward and consider the costs necessary to generate the initial state and the unitary of our phase estimation algorithm respectively. These costs should be seen as the counter part of the costs of sampling from the stationary distribution and sampling a neighbour in a classical random walk.

The **Transition Subroutine** condition has already been discussed indepth in Section 4.3.1, with the addition that we have explicitly bounded the error for most of the edges.

Remark 4.3.2. *For an edge $(u, v) \in \tilde{E}$, we may without loss of generality assume that $v \notin V(G)$. Suppose $i = f_u^{-1}(v)$. Then since we don't actually implement the transition $|u, i\rangle \to |v, j\rangle$ correctly anyway, we can assume that $v = (u, i)$, which is distinct from all vertices in $V(G)$, and so we can consider it an almost isolated vertex with the single backwards neighbour u. We can equivalently think of these as dangling edges, without an endpoint.*

The **Checking Subroutine** also relates to the cost of implementing our unitary. We already saw in Section 3.4.2, that we do not reflect around the star states of marked vertices, so the upper bound on the cost of generating any of the star states must always be upper bounded by this checking cost. To see why this is without loss of generality, suppose the checking cost is some higher value $\mathsf{C} > \mathsf{A}_\star$. Then we can simply put an outgoing edge on each vertex $u \in V_{\mathsf{M}}$ that ends at a new vertex (u, b) that encodes whether $u \in M$ in the bit b. Such an edge can be implemented with transition cost C and we could then analyse the flow whose sinks are in these new vertices.

For the **Positive Condition**, **P1** emphasises that we should not send any flow along the dangling edges from Remark 4.3.2. **P2** and **P3** simply restate the definition of a unit flow, meaning that the flow must be conserved on all vertices that are neither sources nor sinks and that the outgoing flow from the sources must add up to 1. Additionally, **P2** tells us that the flow must also be orthogonal to all additional alternative neighbourhoods that we might have added to $\Psi_\star$. Lastly we will also give some intuition to **P4**. Intuitively, θ should be a σ-M flow, meaning that for all $u \in V_0$, $\theta_u = \sigma(u)$. We don't make this a strict requirement, but this condition means it should hold in some approximate sense.

Implementing the Unitary

Let $\mathcal{A} = \mathrm{span}\{\Psi_{\mathcal{A}}\}$ and $\mathcal{B} = \mathrm{span}\{\Psi_{\mathcal{B}}\}$ (see (4.10)), and let $\Pi_{\mathcal{A}}$ and $\Pi_{\mathcal{B}}$ be the orthogonal projectors onto $\mathcal{A}$ and $\mathcal{B}$. In this section we will prove:

Lemma 4.3.3. *The unitary $U_{\mathcal{AB}} = (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I)$ on $\mathcal{H}$ can be implemented in complexity $O\left(\mathsf{A}_\star + \mathsf{polylog}(\mathsf{T}_{\max})\right)$.*

This will essentially follow from the fact that we can efficiently generate orthonormal bases for each of $\Psi_{\mathcal{A}}$ and $\Psi_{\mathcal{B}}$. For a simple example of how reflecting around a set of states reduces to generating the set, see Claim 4.3.7. We know that we can efficiently generate these orthonormal bases, since

- By the **Star State Generation Subroutine** condition of Theorem 4.3.1, we can generate an orthonormal basis for $\bigcup_{u \in V(G)} \Psi_\star(u)$. Since we can also efficiently check if a vertex is in V_0 or M, we can generate orthonormal bases for $\Psi'_\star = \bigcup_{u \in V(G)} \Psi'_\star(u)$ (see Claim 4.3.4).

- Generating the states $|\psi_t^z\rangle = |z\rangle|t\rangle - U_t|z\rangle|t+1\rangle$ for odd t can be done using $\sum_t |t\rangle\langle t| \otimes U_t$ (see Claim 4.3.5). The same is true for even t (Claim 4.3.6), also including the states $|\psi_\to^{u,i}\rangle = |u, i\rangle|0\rangle - U_0|u, i\rangle|1\rangle$.

- Generating the states $|\psi^{v,j}_{\leftarrow}\rangle = |v,j\rangle(|\mathsf{T}_{u,v}\rangle - |0\rangle)$ can be done efficiently because we can compute $\mathsf{T}_{u,v}$ from (v,j) (see Claim 4.3.7).

There is nothing conceptually new in this proof, and the reader may skip ahead to Section 4.3.3 with no loss of understanding.

Claim 4.3.4. *Let $R_\star = 2\Pi_\star - I$, where $\Pi_\star$ is the orthogonal projector onto $\mathrm{span}\{\Psi'_\star\}$. Then $R_\star$ can be implemented in complexity $O(\mathsf{A}_\star + \log \mathsf{T}_{\max})$.*

Proof. By the **Star State Generation Subroutine** condition of Theorem 4.3.1, we can generate $\Psi_\star$ in cost $\mathsf{A}_\star$, which means (see Definition 4.2.1) that for each $u \in V(G)$, there is an orthonormal basis $\overline{\Psi}(u) = \{|\overline{\psi}_{u,1}\rangle, \ldots, |\overline{\psi}_{u,a_u}\rangle\}$ for $\Psi_\star(u)$, and a unitary $U_\star$ with complexity $\mathsf{A}_\star$, such that for all $u \in V(G)$ and $k \in [a_u - 1]_0$, $U_\star|u,k\rangle = |\overline{\psi}_{u,k}\rangle$. Then for all $u \in V(G) \setminus (V_0 \cup V_{\mathsf{M}})$, $\overline{\Psi}'(u) := \{|\overline{\psi}_{u,1}\rangle|0\rangle, \ldots, |\overline{\psi}_{u,a_u}\rangle|0\rangle\}$ is an orthonormal basis for $\Psi'(u)$ (see (4.6)). For $u \in V_0 \cup V_{\mathsf{M}}$, we have $\Psi'_\star(u) = \{|\psi^{G'}_\star(u)\rangle|0\rangle\}$, so $\overline{\Psi}'(u) = \Psi'_\star(u)$ in those cases.

We will first define a unitary $U'_\star$ that acts, for $u \in V(G)$, $k \in [a_u - 1]_0$, as $U'_\star|u,k\rangle|0\rangle = |\overline{\psi}'_{u,k}\rangle$. We define $U'_\star$ by its implementation. To begin we will append an auxiliary register $|0\rangle_A$ (this will be uncomputed, so that the action described is indeed unitary), and set it to $|1\rangle_A$ if $u \in V_0$. We are assuming we can check if $u \in V_0$ in at most $\mathsf{A}_\star$ complexity. First, controlled on $|0\rangle_A$, we apply $U_\star$, in cost $\mathsf{A}_\star$, to get

$$|u,k\rangle|0\rangle|0\rangle_A \mapsto |\overline{\psi}_{u,k}\rangle|0\rangle|0\rangle_A = |\overline{\psi}'_{u,k}\rangle|0\rangle_A.$$

Next, controlled on $|1\rangle_A$, we implement, on the last register, a single qubit rotation that acts as

$$|0\rangle \mapsto \sqrt{\frac{\mathsf{w}_u}{\mathsf{w}_u + \mathsf{w}_0\sigma(u)}}|1\rangle - \sqrt{\frac{\mathsf{w}_0\sigma(u)}{\mathsf{w}_u + \mathsf{w}_0\sigma(u)}}|0\rangle.$$

Here the weighted degree w_u is taken with respect to the graph G, meaning $\mathsf{w}_u + \mathsf{w}_0\sigma(u) = \mathsf{w}^{G'}_u$. This requires that we can query w_u and $\sigma(u)$ (w_0 is a parameter of the algorithm). Controlled on $|1\rangle$ in the last register (and also still $|1\rangle_A$ in the third), we apply $U_\star$ to get (when $u \in V_0$ we only care about the behaviour for $k = 0$):

$$|u,0\rangle|0\rangle|1\rangle_A \mapsto \left(\sqrt{\frac{\mathsf{w}_u}{\mathsf{w}_u + \mathsf{w}_0\sigma(u)}}|\psi^G_\star(u)\rangle|1\rangle - \sqrt{\frac{\mathsf{w}_0\sigma(u)}{\mathsf{w}_u + \mathsf{w}_0\sigma(u)}}|u,0\rangle|0\rangle\right)|1\rangle_A.$$

Above we have used the fact that when $u \in V_0$, u contains no additional alternative neighbourhoods meaning

$$|\overline{\psi}_{u,0}\rangle = |\psi^G_\star(u)\rangle.$$

To complete the map for the case $u \in V_0$, note that $|\psi^G_\star(u)\rangle$ is supported on $|u,i\rangle$ for $i \neq 0$, so we can uncompute the second register to get

$$\frac{|\psi^G_\star(u)\rangle|0\rangle + \sqrt{\mathsf{w}_0\sigma(u)}|u,0\rangle|0\rangle}{\sqrt{\mathsf{w}_u + \mathsf{w}_0\sigma(u)}}|1\rangle_A = |\psi^{G'}_\star(u)\rangle|0\rangle|1\rangle_A.$$

Since all states still have $|u\rangle$ in the first register, controlled on u, we can uncompute the A register. Thus, we can implement $U'_\star$ in complexity $O(\mathsf{A}_\star)$, and $U'_\star$ maps the subspace

$$\mathcal{L}_\star := \mathrm{span}\{|u,k\rangle|0\rangle : u \in V(G) \setminus M, k \in [a_u - 1]_0\}$$

of $\mathcal{H}$ to the $\mathrm{span}\{\Psi'_\star\}$, meaning $(2\Pi_\star - I) = U'_\star(2\Pi_{\mathcal{L}_\star} - I){U'_\star}^\dagger$.

We complete the proof by describing how to implement $2\Pi_{\mathcal{L}_\star} - I$. Initialise three auxiliary flag qubits, $|0\rangle_{F_1}|0\rangle_{F_2}|0\rangle_{F_3}$. For a computational basis state $|z\rangle|t\rangle$, if $t \neq 0$, flip

F_1 to $|1\rangle_{F_1}$. This check costs $\log \mathsf{T}_{\max}$. If $t = 0$, we can assume that z has the form (u, k), and interpret k as an integer. If $u \in M$, which can be checked in $\mathsf{A}_\star$ by the **Checking Subroutine**, flip F_2 to $|1\rangle_{F_2}$.If $k \geq a_u$, which can be checked in $O(\log d_{\max}) = O(\mathsf{A}_\star)$, flip F_3 to $|1\rangle_{F_3}$. Reflect if either of the flags is set to 1, and then uncompute all three flags. □

Claim 4.3.5. *Let $R_{\mathsf{odd}} = 2\Pi_{\mathsf{odd}} - I$, where Π_{odd} is the orthogonal projector onto the space $\mathrm{span}\{\Psi_t^{u,v} : (u, v) \in \overrightarrow{E}(G), t \in [\mathsf{T}_{u,v} - 1]$ odd$\}$. Then R_{odd} can be implemented in complexity $\mathsf{polylog}(\mathsf{T}_{\max})$.*

Proof. We describe the implementation of a unitary U_{odd} such that

$$\forall (u, v) \in \overrightarrow{E}(G), z \in \mathcal{Z}^0_{u,v}, t \in [\mathsf{T}_{u,v} - 1] \text{ odd}, \; U_{\mathsf{odd}}|z\rangle|t\rangle = \frac{1}{\sqrt{2}}|\psi^t_z\rangle.$$

We begin by decrementing the $|t\rangle$ register, which costs $\log \mathsf{T}_{\max}$. Next we apply an X gate, followed by a Hadamard gate, to the last qubit of $|t - 1\rangle$. If t is odd, $t - 1$ is even and the last qubit is $|0\rangle \overset{\mathsf{HX}}{\mapsto} (|0\rangle - |1\rangle)/\sqrt{2}$, so we get

$$|z\rangle|t\rangle \mapsto |z\rangle|t - 1\rangle \mapsto (|z\rangle|t - 1\rangle - |z\rangle|t\rangle)/\sqrt{2}.$$

Then controlled on the last qubit of $|t\rangle$ being $|1\rangle$ (i.e. on odd parity of t) we apply $\sum_{t=0}^{\mathsf{T}_{\max}-1} |t\rangle\langle t| \otimes U_t$, which can be done in cost $\mathsf{polylog}(\mathsf{T}_{\max})$ by assumption, to obtain $(|z\rangle|t - 1\rangle - U_t|z\rangle|t\rangle)/\sqrt{2}$. Complete the operation by incrementing the $|t\rangle$ register. Thus, U_{odd} maps the subspace:

$$\mathcal{L}_{\mathsf{odd}} := \bigoplus_{(u,v)\in\overrightarrow{E}(G)} \mathrm{span}\{|z\rangle|t\rangle : z \in \mathcal{Z}^0_{u,v}, t \in [\mathsf{T}_{u,v} - 1], \;\; \text{odd}\}$$

of $\mathcal{H}$ to the support of Π_{odd}, and so $R_{\mathsf{odd}} = U_{\mathsf{odd}}(2\Pi_{\mathcal{L}_{\mathsf{odd}}} - I)U^\dagger_{\mathsf{odd}}$.

We complete the proof by describing how to implement $2\Pi_{\mathcal{L}_{\mathsf{odd}}} - I$. For $|z\rangle|t\rangle$, we can check if t is odd in $O(1)$, and if not, set an auxiliary flag F_1. Next, we will ensure that $z \in \mathcal{Z}^0_{u,v}$ for some $(u, v) \in \overrightarrow{E}(G)$, which also ensures that $t \in [\mathsf{T}_{u,v} - 1]_0$, by the structure of $\mathcal{H}$, and if not, set a flag F_2. Reflect if either F_1 or F_2 is set, and then uncompute both of them. □

Claim 4.3.6. *Let $R_{\mathsf{even}} = 2\Pi_{\mathsf{even}} - I$, where Π_{even} is the orthogonal projector onto the span of*

$$\bigcup_{\substack{(u,v)\in\overrightarrow{E}(G),\\ t\in[\mathsf{T}_{u,v}-1]:t \text{ even}}} \Psi_t^{u,v} \cup \{|\psi^{u,i}_{\rightarrow}\rangle : u \in V(G), i \in L^+(u)\}.$$

Then R_{even} can be implemented in complexity $\mathsf{polylog}(\mathsf{T}_{\max})$.

Proof. We describe the implementation of a unitary U_{even} such that for all $(u, v) \in \overrightarrow{E}(G)$ with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$:

$$U_{\mathsf{even}}|u, i\rangle|0\rangle = \frac{1}{\sqrt{2}}(|u, i\rangle|0\rangle - U_0|u, i\rangle|1\rangle) = \frac{1}{\sqrt{2}}|\psi^{u,i}_{\rightarrow}\rangle$$

$$\forall z \in \mathcal{Z}^0_{u,v}, t \in [\mathsf{T}_{u,v} - 1] \text{ even}, \; U_{\mathsf{even}}|z\rangle|t\rangle = \frac{1}{\sqrt{2}}(|z\rangle|t\rangle - U_t|z\rangle|t + 1\rangle) = \frac{1}{\sqrt{2}}|\psi^z_t\rangle.$$

We can implement such a mapping nearly identically to the proof of Claim 4.3.5, except the decrementing of t happens after the Hadamard is applied. Thus, U_{even} maps the subspace:

$$\mathcal{L}_{\mathsf{even}} := \bigoplus_{\substack{u\in V(G),\\ i\in L^+(u)}} \mathrm{span}\{|u, i\rangle|0\rangle\} \oplus \bigoplus_{(u,v)\in\overrightarrow{E}(G)} \mathrm{span}\{|z\rangle|t\rangle : z \in \mathcal{Z}^0_{u,v}, t \in [\mathsf{T}_{u,v} - 1], \text{ even}\}$$

of $\mathcal{H}$ to the support of Π_{even}, and so $R_{\mathsf{even}} = U_{\mathsf{even}}(2\Pi_{\mathcal{L}_{\mathsf{even}}} - I)U_{\mathsf{even}}^{\dagger}$.

We complete the proof by describing how to implement $2\Pi_{\mathcal{L}_{\mathsf{even}}} - I$. For $|z\rangle|t\rangle$, we can check if t is even in $O(1)$ steps, and if not, set an auxiliary flag F_1. Next, we check if $z \in \mathcal{Z}^0_{u,v}$ by checking the subroutine's internal flag, which also ensures that $t \in [\mathsf{T}_{u,v} - 1]_0$, and if not, set a flag F_2. Note that if $t = 0$, z has the form (u, i) for some $i \in L(u)$, by the structure of $\mathcal{H}$, and by the discussion in Section 4.3.1, we can check if $i \in L^+(u)$ in $O(1)$ time, and otherwise, set a flag F_3. Reflect if either F_1, F_2 or F_3 is set, and then uncompute all three flags. □

Claim 4.3.7. *Let $R_{\leftarrow} = 2\Pi_{\leftarrow} - I$, where $\Pi_{\leftarrow}$ is the orthogonal projector onto the span of $\{|\psi^{v,j}_{\leftarrow}\rangle : v \in V(G), j \in L^-(v)\}$. Then $R_{\leftarrow}$ can be implemented in complexity* $\mathsf{polylog}(\mathsf{T}_{\max})$.

Proof. We describe the implementation of a unitary $U_{\leftarrow}$ that acts, for all $v \in V(G)$ and $j \in L^-(v)$, with $u = f_v(j)$, as

$$U_{\leftarrow}|v, j\rangle|0\rangle = \frac{1}{\sqrt{2}}\left(|v, j\rangle|0\rangle - |v, j\rangle|\mathsf{T}_{u,v}\rangle\right) = -\frac{1}{\sqrt{2}}|\psi^{v,j}_{\leftarrow}\rangle.$$

First, append an auxiliary register $|-\rangle_A$. Controlled on this register, compute $\mathsf{T}_{u,v}$ from (v, j), which we can do in $\mathsf{polylog}(\mathsf{T}_{\max})$ basic operations, by the assumptions of Definition 3.5.1, to get

$$|v, j\rangle|0\rangle|-\rangle_A \mapsto |v, j\rangle\left(|0\rangle|0\rangle_A - |\mathsf{T}_{u,v}\rangle|1\rangle_A\right)/\sqrt{2}.$$

We uncompute the A register, by adding 1 conditioned on the time register having a value greater than 0. Thus, $U_{\leftarrow}$ maps the subspace

$$\mathcal{L}_{\leftarrow} := \mathrm{span}\{|v, j\rangle|0\rangle : v \in V(G), j \in L^-(v)\}$$

of $\mathcal{H}$ to the support of $\Pi_{\leftarrow}$, and so $R_{\leftarrow} = U_{\leftarrow}(2\Pi_{\mathcal{L}_{\leftarrow}} - I)U_{\leftarrow}^{\dagger}$.

We complete the proof by describing how to implement $2\Pi_{\mathcal{L}_{\leftarrow}} - I$. Append two auxiliary qubits, $|0\rangle_{F_1}$ and $|0\rangle_{F_2}$. For a computational basis state $|z\rangle|t\rangle$, if $t \neq 0$, which can be checked in $O(\log \mathsf{T}_{\max})$ time, flip F_1 to get $|1\rangle_{F_1}$. By the discussion in Section 4.3.1, we can check if z has the form (v, j) for some $v \in V(G)$ and $j \in L^-(v)$ in $O(1)$ time, and if not, flip F_2 to get $|1\rangle_{F_2}$. Reflect the state if either flag is set to 1, and then uncompute both flags. □

Proof of Lemma 4.3.3. We can see that $\Pi_\star\Pi_{\mathsf{odd}} = 0$, since $\Pi_\star$ is supported on states with 0 in the last register, and Π_{odd} is the span of states with an odd $t \in [\mathsf{T}_{u,v} - 1]$ in the first term, and an even $t \in \{2, \ldots, \mathsf{T}_{u,v}\}$ in the second term. Thus,

$$(2\Pi_{\mathsf{even}} - I)(2\Pi_\star - I) = -(2(\Pi_{\mathsf{even}} + \Pi_\star) - I) = -(2\Pi_{\mathcal{A}} - I),$$

where the last equality is because the support of $\Pi_{\mathcal{A}}$ is the direct sum of the supports of $\Pi_\star$ and Π_{even}, by their definitions. By a similar argument, $(2\Pi_{\mathsf{odd}} - I)(2\Pi_{\leftarrow} - I) = (2\Pi_{\mathcal{B}} - I)$, and thus the result follows from Claim 4.3.4, Claim 4.3.6, Claim 4.3.5 and Claim 4.3.7. □

Positive analysis

Suppose there is a flow θ on G satisfying conditions **P1**-**P5** of Theorem 4.3.1, with corresponding flow state $|\theta\rangle$ (see (4.12)). As stated in the **Positive Condition** of Theorem 4.3.1, the transition subroutine lengths $\mathsf{T}_{u,v}$ induce a network with length (see Definition 3.2.7) where we substitute each edge $(u, v) \in \overrightarrow{E}$ of G with a path from u to v of length $\mathsf{T}_{u,v} + 2$. To construct a positive witness, we will take the flow θ on G and extend this to a flow $\theta^{\mathsf{T}+2}$ on $G^{\mathsf{T}+2}$ by assigning flow $\theta_{u,v}$ to any edge in the path from u to v. To construct

the corresponding flow state $|\theta^{\mathsf{T}+2}\rangle$, define for each $(u,v) \in \overrightarrow{E}(G)$, with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$

$$\begin{aligned} |w_{u,v}^0\rangle &:= |u,i\rangle \\ \forall t \in [\mathsf{T}_{u,v}],\ |w_{u,v}^t\rangle &:= U_{t-1}|w_{u,v}^{t-1}\rangle \\ |w_{u,v}\rangle &:= \sum_{t=0}^{\mathsf{T}_{u,v}} |w_{u,v}^t\rangle|t\rangle + |v,j\rangle|0\rangle. \end{aligned} \tag{4.13}$$

Then $|w_{u,v}\rangle$ is a kind of *history state* [Kit99] for the algorithm on input (u,i). We first show it is almost orthogonal to all transition states, defined in (4.7), (4.8) and (4.9).

Claim 4.3.8. *For all $(u,v) \in \overrightarrow{E}(G)$, letting $j = f_v^{-1}(u)$:*

1. *For all $u' \in V(G)$ and $i' \in L^+(u)$, $\langle\psi_{\rightarrow}^{u',i'}|w_{u,v}\rangle = 0$.*
2. *For all $(u',v') \in \overrightarrow{E}(G)$, $z \in \mathcal{Z}_{u',v'}^0$ and $t \in [\mathsf{T}_{u,v}-1]$, $\langle\psi_t^z|w_{u,v}\rangle = 0$.*
3. *For all $v' \in V(G)$ and $j' \in L^-(u)$, $|\langle\psi_{\leftarrow}^{v',j'}|w_{u,v}\rangle|^2 \le \delta_{(v,j),(v',j')}\epsilon_{u,v}$.*

Proof. **Item 1:** Recalling that $|\psi_{\rightarrow}^{u',i'}\rangle = |u',i'\rangle|0\rangle - U_0|u',i'\rangle|1\rangle$, we have

$$\langle\psi_{\rightarrow}^{u',i'}|w_{u,v}\rangle = \langle u',i'|w_{u,v}^0\rangle - \langle u',i'|U_0^\dagger|w_{u,v}^1\rangle = \langle u',i'|u,i\rangle - \langle u',i'|U_0^\dagger U_0|u,i\rangle = 0.$$

Item 2: Recall that $|\psi_t^z\rangle = |z\rangle|t\rangle - U_t|z\rangle|t+1\rangle$. This is always orthogonal to the last term of $|w_{u,v}\rangle$, since $t > 0$. We also note that if $z \in \mathcal{Z}_{u',v'}$ for $(u',v') \ne (u,v)$, we have $\langle\psi_t^z|w_{u,v}\rangle = 0$, since $|w_{u,v}\rangle$ is only supported on $z \in \mathcal{Z}_{u,v}$. Thus, we can assume $(u,v) = (u',v')$, and so $t \in [\mathsf{T}_{u,v}-1]_0$. Thus:

$$\langle\psi_t^z|w_{u,v}\rangle = \langle z|w_{u,v}^t\rangle - \langle z|U_t^\dagger|w_{u,v}^{t+1}\rangle = \langle z|w_{u,v}^t\rangle - \langle z|U_t^\dagger U_t|w_{u,v}^t\rangle = 0.$$

Item 3: Recall that $|\psi_{\leftarrow}^{v',j'}\rangle = |v',j'\rangle|\mathsf{T}_{u',v'}\rangle - |v',j'\rangle|0\rangle$. Again, if $(v',j') \ne (v,j)$, then $(v',j') \notin \mathcal{Z}_{u,v}$, meaning $\langle\psi_{\leftarrow}^{v',j'}|w_{u,v}\rangle = 0$. So supposing $(v',j') = (v,j)$, we have

$$\langle\psi_{\leftarrow}^{v',j'}|w_{u,v}\rangle = \langle v,j|w_{u,v}^{\mathsf{T}_{u,v}}\rangle - \langle v,j|w_{u,v}^0\rangle - \langle v,j|v,j\rangle = \langle v,j|w_{u,v}^{\mathsf{T}_{u,v}}\rangle - 1,$$

since $|w_{u,v}^0\rangle = |u,i\rangle$, so the middle term is 0. Then, using

$$\left|1 - \langle v,j|w_{u,v}^{\mathsf{T}_{u,v}}\rangle\right|^2 \le \left\||v,j\rangle - |w_{u,v}^{\mathsf{T}_{u,v}}\rangle\right\|^2 = \epsilon_{u,v},$$

by (4.3), we have $|\langle\psi_{\leftarrow}^{v',j'}|w_{u,v}\rangle|^2 \le \epsilon_{u,v}$. □

If not for the extra register containing the time step $|t\rangle$, we have now constructed the flow state $|\theta^{\mathsf{T}+2}\rangle$ of the flow $\theta^{\mathsf{T}+2}$:

$$\frac{1}{\sqrt{\mathcal{E}^{\mathsf{T}+2}(\theta)}} \sum_{(u,v)\in\overrightarrow{E}(G)} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}|w_{u,v}\rangle = \frac{1}{\sqrt{\mathcal{E}^{\mathsf{T}}(\theta) + 2\mathcal{E}(\theta)}} \sum_{(u,v)\in\overrightarrow{E}(G)} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}|w_{u,v}\rangle.$$

Like in Section 3.4, we actually want the flow state to correspond to an s_0-t flow on G'. This can be easily achieved by letting all θ_u of flow in each $u \in V_0$ originate out of s_0. Our positive witness, in the sense of Definition 3.3.5, will be the corresponding unnormalised flow state of this modified new flow (with the extra register containing the time step $|t\rangle$):

$$|w\rangle = \sum_{(u,v)\in\overrightarrow{E}(G)} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}|w_{u,v}\rangle + \sum_{u\in V_0} \frac{\theta_u}{\sqrt{\mathsf{w}_0\sigma(u)}}|u,0\rangle|0\rangle. \tag{4.14}$$

Lemma 4.3.9. *Let* $\mathsf{w}_0 = 1/\mathcal{R}^{\mathsf{T}}$ *and* $c_+ = 6$. *Then* $\frac{\||w\rangle\|^2}{|\langle w|\psi_0\rangle|^2} \leq 6 = c_+$, *and* $\||w\rangle\|^2 \geq \mathcal{E}^{\mathsf{T}}(\theta)$.

Proof. To analyse the positive witness, we compute (referring to (4.11)):

$$\langle\psi_0|w\rangle = \sum_{u\in V_0}\sqrt{\sigma(u)}\langle u,0,0|w\rangle = \sum_{u\in V_0}\sqrt{\sigma(u)}\frac{\theta_u}{\sqrt{\mathsf{w}_0\sigma(u)}} = \frac{1}{\sqrt{\mathsf{w}_0}} = \sqrt{\mathcal{R}^{\mathsf{T}}}, \tag{4.15}$$

by condition **P3** of Theorem 4.3.1. Since this is non-zero, $|w\rangle$ is a positive witness, though it may have some error. To continue, we compute:

$$\||w\rangle\|^2 = \sum_{(u,v)\in\overrightarrow{E}(G)}\frac{\theta_{u,v}^2}{\mathsf{w}_{u,v}}\||w_{u,v}\rangle\|^2 + \sum_{u\in V_0}\frac{\theta_u^2}{\mathsf{w}_0\sigma(u)}. \tag{4.16}$$

To upper bound the first term of (4.16), we have $\||w_{u,v}\rangle\|^2 = \mathsf{T}_{u,v} + 2$, so we have

$$\sum_{(u,v)\in\overrightarrow{E}(G)}\frac{\theta_{u,v}^2}{\mathsf{w}_{u,v}}\||w_{u,v}\rangle\|^2 = \sum_{(u,v)\in\overrightarrow{E}(G)}\frac{\theta_{u,v}^2}{\mathsf{w}_{u,v}}(\mathsf{T}_{u,v}+2) = \mathcal{E}^{\mathsf{T}}(\theta) + 2\mathcal{E}(\theta) \leq 2\mathcal{R}^{\mathsf{T}}, \tag{4.17}$$

by condition **P5** of Theorem 4.3.1, and using $2\mathcal{E}(\theta) \leq \mathcal{E}^{\mathsf{T}}(\theta)$, since each $\mathsf{T}_{u,v} \geq 2$. Note that (4.17) also implies the desired lower bound of $\||w\rangle\|^2 \geq \mathcal{E}^{\mathsf{T}}(\theta)$. To upper bound the second term of (4.16), we have

$$\sum_{u\in V_0}\frac{\theta_u^2}{\mathsf{w}_0\sigma(u)} \leq \frac{2}{\mathsf{w}_0}\sum_{u\in V_0}\frac{\sigma(u)^2+(\theta_u-\sigma(u))^2}{\sigma(u)} = 2\mathcal{R}^{\mathsf{T}}\left(1+\sum_{u\in V_0}\frac{(\theta_u-\sigma(u))^2}{\sigma(u)}\right) \leq 4\mathcal{R}^{\mathsf{T}} \tag{4.18}$$

by condition **P4** of Theorem 4.3.1. Plug (4.17) and (4.18) into (4.16) to get $\||w\rangle\|^2 \leq 6\mathcal{R}^{\mathsf{T}}$, which, with (4.15), completes the proof. □

Next, we analyse the error of $|w\rangle$ as a positive witness, by upper bounding its overlap with the various states in $\Psi_{\mathcal{A}} \cup \Psi_{\mathcal{B}}$. First, we have the following corollary to Claim 4.3.8.

Corollary 4.3.10.
1. *For all* $u \in V(G)$, $i \in L^+(u)$, $\langle\psi_{\rightarrow}^{u,i}|w\rangle = 0$.
2. *For all* $(u,v) \in \overrightarrow{E}(G)$, $z \in \mathcal{Z}_{u,v}^0$ *and* $t \in [\mathsf{T}_{u,v}-1]$, $\langle\psi_t^z|w\rangle = 0$.
3. *For all* $v \in V(G)$ *and* $j \in L^-(v)$, *letting* $u = f_v(j)$, *we have* $|\langle\psi_{\leftarrow}^{v,j}|w\rangle| \leq \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}\sqrt{\epsilon_{u,v}}$.

Next we show that the states in $\Psi'_\star$ are orthogonal to $|w\rangle$.

Claim 4.3.11. *For all* $u' \in V(G)$, *and any* $|\psi_\star\rangle|0\rangle \in \Psi'_\star(u')$, $\langle\psi_\star,0|w\rangle = 0$.

Proof. For any $u' \in V(G)\setminus V_0$ and any $|\psi_\star\rangle|0\rangle \in \Psi'_\star(u')$, the overlap between $|\psi_\star\rangle$ and $|w\rangle$ is only supported on the states $|u',i\rangle$ such that $i \in L(u')$. Hence, we have

$$\langle\psi_\star,0|w\rangle = \sum_{(u,v)\in\overrightarrow{E}(G)}\frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}\langle\psi_\star,0|(|u,f_u^{-1}(v)\rangle|0\rangle + |v,f_v^{-1}(u)\rangle|0\rangle) \propto \langle\psi_\star|\theta\rangle = 0,$$

by condition **P2** of Theorem 4.3.1. If instead $u' \in V_0$, the only state in $\Psi'_\star(u')$ is $|\psi_\star^{G'}(u')\rangle|0\rangle$ (see (4.5)), meaning

$$\langle\psi_\star,0|w\rangle = \langle\psi_\star^{G'}(u'),0|w\rangle \propto \sum_{v\in\Gamma_G^+(u')}\theta_{u',v} - \sum_{u\in\Gamma_G^-(u')}\theta_{u,u'} - \theta_{u'} = 0. \qquad \square$$

We can combine these results in the following lemma:

Lemma 4.3.12. *When $M \neq \emptyset$, $|w\rangle$ as defined in (4.14) is an $\epsilon/2$-positive witness (see Definition 3.3.5).*

Proof. Note that $|w\rangle$ is only defined when $M \neq \emptyset$, as it is constructed with a flow from V_0 to M. For $|w\rangle$ to be a positive witness, we require that $\langle w|\psi_0\rangle \neq 0$, which follows from (4.15). All that remains is to show that $\|\Pi_{\mathcal{A}}|w\rangle\|^2$ and $\|\Pi_{\mathcal{B}}|w\rangle\|^2$ are both at most $\frac{\epsilon}{2}\||w\rangle\|^2$. By Claim 4.3.8 and Claim 4.3.11, we have $\|\Pi_{\mathcal{A}}|w\rangle\|^2 = 0$ and

$$\sum_{v\in V(G), j\in L^-(v)} \frac{|\langle \psi_{\leftarrow}^{v,j}|w\rangle|^2}{\left\||\psi_{\leftarrow}^{v,j}\rangle\right\|^2} \leq \sum_{v\in V(G), j\in L^-(v)} \frac{\theta_{v,f_v(j)}^2 \epsilon_{f_v(j),v}/\mathsf{w}_{v,f_v(j)}}{2}.$$

Since $\theta_e = 0$ for all $e \in \tilde{E}$ (condition **P1** of Theorem 4.3.1) and for all $e \in \overrightarrow{E}(G) \setminus \tilde{E}$, $\epsilon_{u,v} \leq \epsilon$ (condition **TS1** of Theorem 4.3.1), we can continue:

$$\|\Pi_{\mathcal{B}}|w\rangle\|^2 \leq \frac{1}{2} \sum_{(u,v)\in \overrightarrow{E}(G)\setminus\tilde{E}} \frac{\theta_{u,v}^2 \epsilon_{u,v}}{\mathsf{w}_{u,v}} \leq \frac{\epsilon}{2}\mathcal{E}(\theta) < \frac{\epsilon}{2}\mathcal{E}^{\mathsf{T}}(\theta) \leq \frac{\epsilon}{2}\||w\rangle\|^2$$

by Lemma 4.3.9. We have used the fact that the energy of the flow in G (see Definition 3.2.3), $\mathcal{E}(\theta)$, is at most the energy of that flow extended to the graph G^{T} in which we replace the edges by paths of positive lengths determined by T (see Definition 3.2.7). □

Negative analysis

In this section, we will define a negative witness, which is some $|w_{\mathcal{A}}\rangle, |w_{\mathcal{B}}\rangle \in \mathcal{H}$, such that $|\psi_0\rangle = |w_{\mathcal{A}}\rangle + |w_{\mathcal{B}}\rangle$ and $|w_{\mathcal{A}}\rangle$ (resp. $|w_{\mathcal{B}}\rangle$) is almost in $\mathcal{A}$ (resp. $\mathcal{B}$) (see Definition 3.3.2). We first define, for all $(u,v) \in \overrightarrow{E}(G)$ with $i = f_u^{-1}(v)$,

$$\begin{aligned} |w_{u,v}^{\mathcal{A}}\rangle &= \sum_{t\in[\mathsf{T}_{u,v}-1]:t \text{ odd}} \sum_{z\in\mathcal{Z}_{u,v}^0} \langle z|w_{u,v}^t\rangle|\psi_t^z\rangle \in \mathcal{A} \\ |w_{u,v}^{\mathcal{B}}\rangle &= |\psi_{\rightarrow}^{u,i}\rangle + \sum_{t\in[\mathsf{T}_{u,v}-1]:t \text{ even}} \sum_{z\in\mathcal{Z}_{u,v}^0} \langle z|w_{u,v}^t\rangle|\psi_t^z\rangle \in \mathcal{B}, \end{aligned} \tag{4.19}$$

where $|w_{u,v}^t\rangle$ is defined in (4.13).

Lemma 4.3.13. *For all $(u,v) \in \overrightarrow{E}(G)$ with $i = f_u^{-1}(v)$,*

$$|w_{u,v}^{\mathcal{A}}\rangle + |w_{u,v}^{\mathcal{B}}\rangle = |u,i\rangle|0\rangle - |w_{u,v}^{\mathsf{T}_{u,v}}\rangle|\mathsf{T}_{u,v}\rangle.$$

Proof. Below we use the fact that for $t \in [\mathsf{T}_{u,v}-1]_0$, $|w_{u,v}^t\rangle \in \operatorname{span}\{|z\rangle : z \in \mathcal{Z}_{u,v}^0\}$ (see

Section 4.3.1), and that $|w^0_{u,v}\rangle = |u,i\rangle$ (see (4.13)).

$$\begin{aligned}
|w^{\mathcal{A}}_{u,v}\rangle + |w^{\mathcal{B}}_{u,v}\rangle &= |\psi^{u,i}_{\rightarrow}\rangle + \sum_{t\in[\mathsf{T}_{u,v}-1]}\sum_{z\in\mathcal{Z}^0_{u,v}} \langle z|w^t_{u,v}\rangle|\psi^z_t\rangle \\
&= |\psi^{u,i}_{\rightarrow}\rangle + \sum_{t=1}^{\mathsf{T}_{u,v}-1}\sum_{z\in\mathcal{Z}^0_{u,v}} \langle z|w^t_{u,v}\rangle|z,t\rangle - \sum_{t=1}^{\mathsf{T}_{u,v}-1}\sum_{z\in\mathcal{Z}^0_{u,v}} \langle z|w^t_{u,v}\rangle U_t|z\rangle|t+1\rangle \qquad \text{see (4.8)} \\
&= |\psi^{u,i}_{\rightarrow}\rangle + \sum_{t=1}^{\mathsf{T}_{u,v}-1} |w^t_{u,v}\rangle|t\rangle - \sum_{t=1}^{\mathsf{T}_{u,v}-1} U_t|w^t_{u,v}\rangle|t+1\rangle \\
&= |\psi^{u,i}_{\rightarrow}\rangle + \sum_{t=1}^{\mathsf{T}_{u,v}-1} |w^t_{u,v}\rangle|t\rangle - \sum_{t=1}^{\mathsf{T}_{u,v}-1} |w^{t+1}_{u,v}\rangle|t+1\rangle \qquad \text{see (4.13)} \\
&= |u,i\rangle|0\rangle - U_0|u,i\rangle|1\rangle + |w^1_{u,v}\rangle|1\rangle - |w^{\mathsf{T}_{u,v}}_{u,v}\rangle|\mathsf{T}_{u,v}\rangle \qquad \text{see (4.7)} \\
&= |u,i\rangle|0\rangle - |w^{\mathsf{T}_{u,v}}_{u,v}\rangle|\mathsf{T}_{u,v}\rangle,
\end{aligned}$$

since $|w^1_{u,v}\rangle = U_0|u,i\rangle$ (see (4.13)). □

For $v \in V(G)$ and $j \in L^-(v)$, with $u = f_v(j)$, define

$$|\tilde{\psi}^{v,j}_{\leftarrow}\rangle := |w^{\mathsf{T}_{u,v}}_{u,v}\rangle|\mathsf{T}_{u,v}\rangle - |v,j\rangle|0\rangle, \tag{4.20}$$

which would be equal to $|\psi^{v,j}_{\leftarrow}\rangle$ if not for the possible errors in our transition subroutines. We use this state to define our negative witness:

$$\begin{aligned}
|w_{\mathcal{A}}\rangle &= -\frac{1}{\sqrt{\mathsf{w}_0}}\sum_{u\in V(G)}\sqrt{\mathsf{w}'_u}|\psi^{G'}_{\star}(u)\rangle|0\rangle + \frac{1}{\sqrt{\mathsf{w}_0}}\sum_{(u,v)\in\vec{E}(G)}\sqrt{\mathsf{w}_{u,v}}|w^{\mathcal{A}}_{u,v}\rangle \in \mathcal{A} \\
|w_{\mathcal{B}}\rangle &= \frac{1}{\sqrt{\mathsf{w}_0}}\sum_{(u,v)\in\vec{E}(G)}\sqrt{\mathsf{w}_{u,v}}\left(|w^{\mathcal{B}}_{u,v}\rangle + |\tilde{\psi}^{v,f_v^{-1}(u)}_{\leftarrow}\rangle\right).
\end{aligned}$$

Here the weighted degree w'_u of u is taken with respect to the graph G', which we emphasise with the prime in the superscript.

We first show that this is indeed a negative witness in Lemma 4.3.14 and then analyse its error and complexity in Lemma 4.3.15.

Lemma 4.3.14. *Let $|\psi_0\rangle$ be as in (4.11). Then if $M = \emptyset$, $|w_{\mathcal{A}}\rangle + |w_{\mathcal{B}}\rangle = |\psi_0\rangle$.*

Proof. We have

$$\begin{aligned}
&-\sqrt{\mathsf{w}_0}\left(|w_{\mathcal{A}}\rangle + |w_{\mathcal{B}}\rangle\right) \\
&= \sum_{u\in V(G)}\sqrt{\mathsf{w}'_u}|\psi^{G'}_{\star}(u)\rangle|0\rangle - \sum_{(u,v)\in\vec{E}(G)}\sqrt{\mathsf{w}_{u,v}}(|w^{\mathcal{A}}_{u,v}\rangle + |w^{\mathcal{B}}_{u,v}\rangle + |\tilde{\psi}^{v,f_v^{-1}(u)}_{\leftarrow}\rangle).
\end{aligned} \tag{4.21}$$

Letting $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$, we have

$$\begin{aligned}
|w^{\mathcal{A}}_{u,v}\rangle + |w^{\mathcal{B}}_{u,v}\rangle + |\tilde{\psi}^{v,j}_{\leftarrow}\rangle &= |u,i\rangle|0\rangle - |w^{\mathsf{T}_{u,v}}_{u,v}\rangle|\mathsf{T}_{u,v}\rangle + |\tilde{\psi}^{v,j}_{\leftarrow}\rangle \quad \text{by Lemma 4.3.13} \\
&= |u,i\rangle|0\rangle - |v,j\rangle|0\rangle \qquad \text{by (4.20).}
\end{aligned} \tag{4.22}$$

Next we recall from (4.5) that for $u \in V(G)\setminus M = V(G)$ (since $M = \emptyset$), we have, letting $\delta_{u,V_0} = 1$ iff $u \in V_0$:

$$\begin{aligned}
\sqrt{\mathsf{w}'_u}|\psi^{G'}_{\star}(u)\rangle &= \sum_{i\in L^+(u)}\sqrt{\mathsf{w}_{u,i}}|u,i\rangle - \sum_{j\in L^-(u)}\sqrt{\mathsf{w}_{u,j}}|u,j\rangle - \delta_{u,V_0}\sqrt{\mathsf{w}_0\sigma(u)}|u,0\rangle \\
\sum_{u\in V(G)}\sqrt{\mathsf{w}'_u}|\psi^{G'}_{\star}(u)\rangle &= \sum_{(u,v)\in\vec{E}(G)}\sqrt{\mathsf{w}_{u,v}}\left(|u,f_u^{-1}(v)\rangle - |v,f_v^{-1}(u)\rangle\right) - \sum_{u\in V_0}\sqrt{\mathsf{w}_0\sigma(u)}|u,0\rangle.
\end{aligned} \tag{4.23}$$

Plugging (4.22) and (4.23) back into (4.21), we get

$$-\sqrt{\mathsf{w}_0}\left(|w_{\mathcal{A}}\rangle + |w_{\mathcal{B}}\rangle\right) = -\sum_{u\in V_0}\sqrt{\mathsf{w}_0\sigma(u)}|u,0\rangle|0\rangle = -\sqrt{\mathsf{w}_0}|\psi_0\rangle. \qquad \square$$

Lemma 4.3.15. *Let* $\mathsf{w}_0 = 1/\mathcal{R}^{\mathsf{T}}$*, and* $\delta' = \epsilon\mathcal{R}^{\mathsf{T}}\mathcal{W} + 4\mathcal{R}^{\mathsf{T}}\widetilde{\mathcal{W}}$*. Then* $|w_{\mathcal{A}}\rangle, |w_{\mathcal{B}}\rangle$ *is a* δ'*-negative witness (see Definition 3.3.2), and*

$$\||w_{\mathcal{A}}\rangle\|^2 \le 2\mathcal{R}^{\mathsf{T}}\mathcal{W}^{\mathsf{T}} + 1.$$

Proof. By construction, $|w_{\mathcal{A}}\rangle \in \mathcal{A}$, and the only part of $|w_{\mathcal{B}}\rangle$ that is not made up of states in $\Psi_{\mathcal{B}}$ (which are in $\mathcal{B} = \operatorname{span}\{\Psi_{\mathcal{B}}\}$) are the $|\tilde{\psi}^{v,j}_{\leftarrow}\rangle$ parts. Since $(I-\Pi_{\mathcal{B}})|\psi^{v,j}_{\leftarrow}\rangle = 0$ for all (v,j), we have

$$\begin{aligned}
(I-\Pi_{\mathcal{B}})|w_{\mathcal{B}}\rangle &= \frac{1}{\sqrt{\mathsf{w}_0}}\sum_{v\in V(G), j\in L^-(v)}\sqrt{\mathsf{w}_{v,j}}(I-\Pi_{\mathcal{B}})|\tilde{\psi}^{v,j}_{\leftarrow}\rangle \\
&= \frac{1}{\sqrt{\mathsf{w}_0}}\sum_{v\in V(G), j\in L^-(v)}\sqrt{\mathsf{w}_{v,j}}(I-\Pi_{\mathcal{B}})(|\tilde{\psi}^{v,j}_{\leftarrow}\rangle - |\psi^{vj}_{\leftarrow}\rangle) \\
&= \frac{1}{\sqrt{\mathsf{w}_0}}\sum_{(u,v)\in\overrightarrow{E}(G)}\sqrt{\mathsf{w}_{u,v}}(I-\Pi_{\mathcal{B}})(|w^{\mathsf{T}_{u,v}}_{u,v}\rangle|\mathsf{T}_{u,v}\rangle - |v, f_v^{-1}(u)\rangle|\mathsf{T}_{u,v}\rangle)
\end{aligned}$$

by (4.20) and (4.9). Then by (4.13) and (4.3) we have

$$\left\||w^{\mathsf{T}_{u,v}}_{u,v}\rangle - |v, f_v^{-1}(u)\rangle\right\|^2 = \epsilon_{u,v}.$$

Furthermore, the terms $|w^{\mathsf{T}_{u,v}}_{u,v}\rangle - |v, f_v^{-1}(u)\rangle \in \operatorname{span}\{|z\rangle : z \in \mathcal{Z}_{u,v}\}$ for different $(u,v) \in \overrightarrow{E}(G)$ are pairwise orthogonal. Thus:

$$\begin{aligned}
\|(I-\Pi_{\mathcal{B}})|w_{\mathcal{B}}\rangle\|^2 &\le \frac{1}{\mathsf{w}_0}\sum_{(u,v)\in\overrightarrow{E}(G)}\mathsf{w}_{u,v}\epsilon_{u,v} \\
&\le \mathcal{R}^{\mathsf{T}}\left(\sum_{(u,v)\in\overrightarrow{E}(G)\setminus\tilde{E}}\mathsf{w}_{u,v}\epsilon + 4\sum_{(u,v)\in\tilde{E}}\mathsf{w}_{u,v}\right) \le \mathcal{R}^{\mathsf{T}}\left(\epsilon\mathcal{W} + 4\widetilde{\mathcal{W}}\right),
\end{aligned}$$

where we used $\mathsf{w}_0 = 1/\mathcal{R}^{\mathsf{T}}$, the trivial upper bound $\epsilon_{u,v} \le 4$ when $(u,v) \in \tilde{E}$, and conditions **TS1** and **TS2** of Theorem 4.3.1.

To complete the proof, we give an upper bound on $\||w_{\mathcal{A}}\rangle\|^2$. We first note:

$$\||w_{\mathcal{A}}\rangle\|^2 = \frac{1}{\mathsf{w}_0}\sum_{u\in V(G)}\mathsf{w}'_u + \frac{1}{\mathsf{w}_0}\sum_{(u,v)\in\overrightarrow{E}(G)}\mathsf{w}_{u,v}\left\||w^{\mathcal{A}}_{u,v}\rangle\right\|^2.$$

Then we can compute, for any $u \in V(G)$, letting $\delta_{u,V_0} = 1$ iff $u \in V_0$,

$$\mathsf{w}'_u = \sum_{v\in\Gamma(u)}\mathsf{w}_{u,v} + \delta_{u,V_0}\mathsf{w}_0\sigma(u)$$

and for any $(u,v) \in \overrightarrow{E}(G)$, since for all $t \in [\mathsf{T}_{u,v}-1]_0$, $\sum_{z\in\mathcal{Z}^0_{u,v}}|\langle z|w^t_{u,v}\rangle|^2 = \left\||w^t_{u,v}\rangle\right\|^2 = 1$,

$$\left\||w^{\mathcal{A}}_{u,v}\rangle\right\|^2 = \sum_{t\in[\mathsf{T}_{u,v}-1]_0 : t \text{ odd}}\sum_{z\in\mathcal{Z}^0_{u,v}}|\langle z|w^t_{u,v}\rangle|^2\,\||\psi^z_t\rangle\|^2 = \left\lfloor\frac{\mathsf{T}_{u,v}}{2}\right\rfloor\cdot 2 \le \mathsf{T}_{u,v}.$$

Phase estimation	Multidimensional quantum walk
$\mathcal{H}$	$\mathrm{span}\{\lvert u,i\rangle\lvert 0\rangle : u \in V(G), i \in L^+(u) \cup \{0\}\}$ $\oplus \mathrm{span}\{\lvert v,j\rangle\lvert 0\rangle : v \in V(G), j \in L^-(v)\}$ $\oplus \bigoplus_{(u,v)\in\overrightarrow{E}(G)} \big(\mathrm{span}\{\lvert z\rangle\lvert t\rangle : z \in \mathcal{Z}^0_{u,v}, t \in [\mathsf{T}_{u,v}-1]\}$ $\cup \{\lvert z\rangle\lvert \mathsf{T}_{u,v}\rangle : z \in \mathcal{Z}^1_{u,v}\}\big)$
$\lvert\psi_0\rangle$	$\sum_{u\in V_0} \sqrt{\sigma(u)}\lvert u,0\rangle\lvert 0\rangle$.
$\Psi_{\mathcal{A}}$	$\Psi'_\star \cup \bigcup_{(u,v)\in\overrightarrow{E}(G)} \bigcup_{\substack{t=1:\\ t \text{ odd}}}^{\mathsf{T}_{u,v}-1} \Psi_t^{u,v}$ where $\Psi'_\star$ is defined as in (4.6)
$\Psi_{\mathcal{B}}$	$\bigcup_{u\in V(G)} \{\lvert\psi^{u,i}_{\rightarrow}\rangle : i \in L^+(u)\} \cup \{\lvert\psi^{u,j}_{\leftarrow}\rangle : j \in L^-(v)\}$ $\cup \bigcup_{(u,v)\in\overrightarrow{E}(G)} \bigcup_{\substack{t=1:\\ t \text{ even}}}^{\mathsf{T}_{u,v}-1} \Psi_t^{u,v}$
C_-	$2\mathcal{R}^{\mathsf{T}}\mathcal{W}^{\mathsf{T}} + 1$
$\lvert w\rangle$	$\sum_{(u,v)\in\overrightarrow{E}(G)} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}\lvert w_{u,v}\rangle + \sum_{u\in V_0} \frac{\theta_u}{\sqrt{\mathsf{w}_0\sigma(u)}}\lvert u,0\rangle\lvert 0\rangle$ for any unit flow θ satisfying **P1**-**P5**
$\lvert w_{\mathcal{A}}\rangle$	$-\sqrt{\mathcal{R}^{\mathsf{T}}} \sum_{u\in V(G)} \sqrt{\mathsf{w}'_u}\lvert\psi_\star^{G'}(u)\rangle\lvert 0\rangle + \sqrt{\mathcal{R}^{\mathsf{T}}} \sum_{(u,v)\in\overrightarrow{E}(G)} \sqrt{\mathsf{w}_{u,v}}\lvert w^{\mathcal{A}}_{u,v}\rangle$
$\lvert w_{\mathcal{B}}\rangle$	$\sqrt{\mathcal{R}^{\mathsf{T}}} \sum_{(u,v)\in\overrightarrow{E}(G)} \sqrt{\mathsf{w}_{u,v}} \left(\lvert w^{\mathcal{B}}_{u,v}\rangle + \lvert\tilde{\psi}^{v,f_v^{-1}(u)}_{\leftarrow}\rangle\right)$

Table 4.1: A summary of how we initialise the phase estimation parameters of Theorem 3.3.8 in the proof of Theorem 4.3.1.

Thus:

$$\begin{aligned}\|\lvert w_{\mathcal{A}}\rangle\|^2 &\leq \frac{1}{\mathsf{w}_0}\sum_{u\in V(G)}\sum_{v\in\Gamma(u)} \mathsf{w}_{u,v} + \frac{1}{\mathsf{w}_0}\sum_{u\in V_0} \mathsf{w}_0\sigma(u) + \frac{1}{\mathsf{w}_0}\sum_{(u,v)\in\overrightarrow{E}(G)} \mathsf{w}_{u,v}\mathsf{T}_{u,v} \\ &= \frac{1}{\mathsf{w}_0}\mathcal{W}(G) + 1 + \frac{1}{\mathsf{w}_0}\mathcal{W}^{\mathsf{T}}(G).\end{aligned}$$

Note that $\mathcal{W}(G)$ is always less than $\mathcal{W}^{\mathsf{T}}(G)$ (see Definition 3.2.7). We thus complete the proof by substituting $\mathsf{w}_0 = 1/\mathcal{R}^{\mathsf{T}}$ and using the **Negative Condition** of Theorem 4.3.1 that $\mathcal{W}^{\mathsf{T}}(G) \leq \mathcal{W}^{\mathsf{T}}$. □

We summarise all these parameters in Table 4.1and encourage the reader to compare these with the paremeters from Table 3.1.

Conclusion of proof of Theorem 4.3.1

We now give the proof of Theorem 4.3.1, by appealing to Theorem 3.3.8, using $\lvert\psi_0\rangle$ as defined in (4.11), and $\Psi_{\mathcal{A}}, \Psi_{\mathcal{B}}$ as defined in (4.10).

By the **Setup Subroutine** condition of Theorem 4.3.1, we can generate $\lvert\sigma\rangle$ in cost S. It follows that we can generate $\lvert\psi_0\rangle = -\lvert\sigma\rangle\lvert 0\rangle\lvert 0\rangle$ in cost $\mathsf{S}' = \mathsf{S} + \log \mathsf{T}_{\max}$, since the last register is $\log \mathsf{T}_{\max}$ qubits. By Lemma 4.3.3, we can implement $U_{\mathcal{AB}}$ in cost $\mathsf{A}_\star + \mathsf{polylog}(\mathsf{T}_{\max})$.

We use $c_+ = 6$, so $c_+ \in [1, 50]$, as desired. We use

$$\mathcal{C}_- = 2\mathcal{R}^{\mathsf{T}}\mathcal{W}^{\mathsf{T}} + 1, \quad \delta = \frac{\epsilon}{2} \quad \text{and} \quad \delta' = \epsilon\mathcal{R}^{\mathsf{T}}\mathcal{W} + 4\mathcal{R}^{\mathsf{T}}\widetilde{\mathcal{W}}.$$

To apply Theorem 3.3.8, we require that $\delta \leq \frac{1}{(8c_+)^3\pi^8\mathcal{C}_-}$, which follows, for sufficiently large $|x|$, from condition **TS1** of Theorem 4.3.1:

$$\delta = \frac{\epsilon}{2} = o\left(\frac{1}{\mathcal{R}^{\mathsf{T}}\mathcal{W}^{\mathsf{T}}}\right).$$

We also require that $\delta' \leq \frac{3}{4}\frac{1}{\pi^4 c_+} = \frac{1}{8\pi^4}$. The bound on ϵ implies that $\epsilon\mathcal{R}^{\mathsf{T}}\mathcal{W} = o(1)$, since $\mathcal{W} \leq \mathcal{W}^{\mathsf{T}}$. The bound $\widetilde{\mathcal{W}} = o(1/\mathcal{R}^{\mathsf{T}})$ from **TS2** of Theorem 4.3.1 implies that $4\mathcal{R}^{\mathsf{T}}\widetilde{\mathcal{W}} = o(1)$.

Together these ensure that $\delta' = o(1)$. We verify the remaining conditions of Theorem 3.3.8 as follows.

Positive Condition: By Lemma 4.3.9 and Lemma 4.3.12, if $M \neq \emptyset$, there is a δ-positive witness $|w\rangle$ such that $\frac{|\langle w|\psi_0\rangle|^2}{\||w\rangle\|^2} \geq \frac{1}{c_+} = \frac{1}{6}$.

Negative Condition: By Lemma 4.3.15, if $M = \emptyset$, there is a δ'-negative witness with $\||w_{\mathcal{A}}\rangle\|^2 \leq \mathcal{C}_-$.

Thus, the algorithm described in Theorem 3.3.8 distinguishes between the cases $M \neq \emptyset$ and $M = \emptyset$ with bounded error in complexity:

$$O\left(\mathsf{S} + \log \mathsf{T}_{\max} + \sqrt{\mathcal{C}_-}\left(\mathsf{A}_\star + \mathsf{polylog}(\mathsf{T}_{\max})\right)\right) = O\left(\mathsf{S} + \sqrt{\mathcal{R}^{\mathsf{T}}\mathcal{W}^{\mathsf{T}}}\left(\mathsf{A}_\star + \mathsf{polylog}(\mathsf{T}_{\max})\right)\right)$$

which completes the proof of Theorem 4.3.1.

4.4 Welded trees

A straightforward application of our technique is to the *welded trees* problem, first introduced in [CCD+03], illustrating the power of the framework to achieve exponential speedups over classical algorithms. This application also serves as a pedagogic demonstration of the alternative neighbourhoods technique, as it does not make use of a non-trivial edge transition subroutine, and so the resulting algorithm is in that sense rather simple. Although it would be possible to apply our framework without looking "under the hood" at the underlying algorithm, to give intuition about the framework, we instead describe and analyse the full algorithm explicitly, proving our upper bound without appealing to Theorem 4.3.1. Since we do not make use of the edge transitions, we will in fact use the initialisation from Section 3.4.

4.4.1 The welded trees problem

In the welded trees problem, the input is a graph G with $2^{n+2} - 2$ vertices from the set $\{0,1\}^{2n}$, consisting of two full binary trees of depth n (the 2^n leaves are at edge-distance n from the root), which we will refer to as the left and right trees, with additional edges connecting the leaves of one tree to another. Specifically, we assume there are two disjoint perfect matchings from the leaves of the left tree to the leaves of the right tree. Every vertex of this graph has degree 3 except for the roots of the two trees, which we denote by s and t. The graph's structure is shown in Figure 4.8.

We are promised that $s = 0^{2n}$ is the root of the left tree, but other than s, it is difficult to even find a vertex in the graph, since less than a 2^{-n+2} fraction of strings in $\{0,1\}^{2n}$ labels an actual vertex. We assume we have access to an oracle O_G that tells us the neighbours of any vertex. That is, for any string $\sigma \in \{0,1\}^{2n}$, we can query $O_G(\sigma)$ to learn either $\perp$, indicating it is not a vertex label, or a list of three neighbours (or in case of s and t, only two neighbours).

Problem 4.4.1 (Welded tree problem)**.** *We are given an adjacency list oracle O_G for the welded trees graph G of depth n, the goal is to output the $2n$-bit string associated to the root t.*

We assume we can identify t when we see it, for example by querying it to see that it only has two neighbours. Classically, this problem requires $2^{\Omega(n)}$ queries [CCD+03], which is intuitively because the problem is set up to ensure that the only thing a classical algorithm can do is a random walk on G, starting from s. The hitting time from s to t is $2^{\Omega(n)}$ because a walker is always twice as likely to move towards the centre of the graph than away from it, and so a walker starting at s will quickly end up in the centre of the graph, but then will be stuck there for a long time. On the other hand a quantum algorithm can solve this problem in poly(n) time [CCD+03], with the best known upper bound being $O(n^{1.5}\log n)$ queries [AC21]. In fact, there is even a deterministic quantum algorithm that solves this problem with success probability 1 in $O(n^{1.5}\log n)$ queries [LLL24]. We show how to solve this problem in our new framework, with $O(n)$ queries and $O(n^2)$ time. Specifically, in the remainder of this section we show:

Theorem 4.4.2. *Let $g : \{0,1\}^{2n} \to \{0,1\}$ be any function. Then there is a quantum algorithm that, given an oracle O_G for a welded trees graph G as above, decides if $g(t) = 1$ with bounded error in $O(n)$ queries to O_G. If g can be computed in $O(n)$ complexity, then the time complexity of this algorithm is $O(n^2)$.*

From this it immediately follows that we can solve the welded trees problem with $2n$ applications of this algorithm, letting $g(t) = t_i$ – the i-th bit of t – for $i \in [2n]$. However, we can also do slightly better by composing it with the Bernstein-Vazirani algorithm, which recovers a string t in a single quantum query to an oracle that computes $|z\rangle \mapsto (-1)^{z\cdot t}|z\rangle$ for any string $z \in \{0,1\}^{2n}$.

Corollary 4.4.3. *There is a quantum algorithm that can solve the welded trees problem in $O(n)$ queries and $O(n^2)$ time.*

Proof. For any $z \in \{0,1\}^{2n}$, define $g_z(t) = z \cdot t = \sum_{i=1}^{2n} z_i t_i \mod 2$. Clearly g_z can be computed in complexity $O(n)$. To compute $g_z(t)$, we simply run the algorithm from Theorem 4.4.2. The Bernstein-Vazirani algorithm [BV97] outputs t using a single such query, and $O(n)$ additional gates. □

Previous quantum algorithms for this problem are quantum walk algorithms in the sense that they construct a Hamiltonian based on the structure of the graph and simulate it, or they exponentially reduce the dimension of the Hilbert space on which the quantum walk takes place due to the inherent symmetry of the problem. These techniques have not been replicated for many other problems, unlike quantum walk search algorithms described in Section 3.1. Through this application, we hope that our new quantum walk search framework bridges the gap between a general and easily applied technique (quantum walk search algorithms) and exponential speedups.

4.4.2 G as a weighted network

We assume for simplicity that n is even. This assumption is without loss of generality and greatly simplifies notation. For the skeptical reader, we will consider the case where n is odd in Chapter 6. We partition $V(G)$ into $V_0 \cup V_1 \cup \cdots \cup V_{2n+1}$, where V_k is the set of vertices at distance k from s, so $V_0 = \{s\}$, and $V_{2n+1} = \{t\}$. We first prove Theorem 4.4.2 under the assumption that it is possible to check, for any vertex, whether it is in $V_{\text{even}} := V_0 \cup V_2 \cup \cdots \cup V_{2n}$, or $V_{\text{odd}} := V_1 \cup V_3 \cup \cdots \cup V_{2n+1}$. At the end of this section, we will explain how to remove this assumption. Define $M = \{t\}$ if $g(t) = 1$ and otherwise $M = \emptyset$.

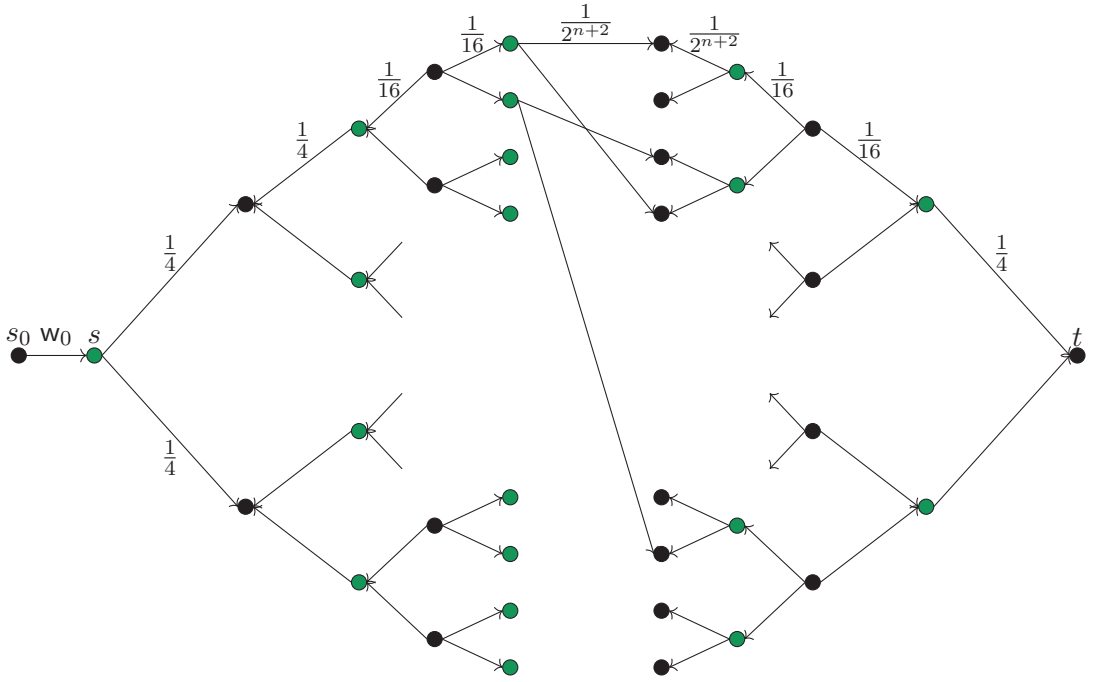


Figure 4.8: The weights of the graph G' (obtained from adding s_0 to G), and default edge directions. The vertices in V_{even} are coloured.

For $k \in [2n+1]$, define

$$E_k = \{(u,v) \in V_{k-1} \times V_k : (u,v) \in E(G)\}$$
$$\text{so } |E_k| = \begin{cases} 2^k & \text{if } k \in [n+1] \\ 2^{2n+2-k} & \text{if } k \in \{n+1, \dots, 2n+1\}. \end{cases} \tag{4.24}$$

We define the set of directed edges as follows (see Definition 3.2.1):

$$\overrightarrow{E}(G) = \bigcup_{\substack{k \in [2n+1]: \\ k \bmod 4 \in \{0,1\}}} \{(u,v) : (u,v) \in E_k\} \cup \bigcup_{\substack{k \in [2n+1]: \\ k \bmod 4 \in \{2,3\}}} \{(u,v) : (v,u) \in E_k\}. \tag{4.25}$$

Note that $E_k \subset \overrightarrow{E}(G)$ only holds when $k \mod 4 \in \{0,1\}$, so we do not always set the default directions left to right. At the moment it is not clear why we set the directions this way, but one thing this accomplishes is that the direction of edges switches at every layer of V_{odd}. Figure 4.8 illustrates how the directions of the edges change layer by layer.

We now assign weights to all edges in G. We assign all edges in E_k the same weight, w_k, defined:

$$\mathsf{w}_k = \begin{cases} 2^{-2\lceil k/2 \rceil} & \text{if } k \in [n] \\ 2^{-2(n+2-\lceil k/2 \rceil)} & \text{if } k \in \{n+1, \dots, 2n+1\}. \end{cases} \tag{4.26}$$

It should be somewhat clear why this might be a useful weighting: we have increased the probability of moving away from the centre. Finally, we add a vertex s_0 connected to s by an edge of weight w_0 and call this resulting graph G'. We remark that we do not need to account for s_0 explicitly if we just want to appeal directly to Theorem 4.3.1, but we are going to explicitly construct an algorithm for the sake of exemplification.

4.4.3 The phase estimation parameters

Hilbert space: Theorem 4.3.1 assumes we label the outgoing edges from a vertex u by indices from some set $L(u)$, and then implement a map $|u,i\rangle \mapsto |v,j\rangle$ for any $i \in L(u)$ (see

also Definition 3.2.13). Here, since we assume we can simply query the set of the three neighbours of u, $\Gamma(u) = \{v_1, v_2, v_3\}$, in unit time, we can let $L(u) = \Gamma(u)$. In that case, the map $|u, i\rangle \mapsto |v, j\rangle$ is actually just $|u, v\rangle \mapsto |v, u\rangle$, which can be accomplished by swapping the two registers, and there is no need for more complex transition subroutines. The decomposition of $L(u)$ into $L^+(u) = \Gamma^+(u)$ and $L^-(u) = \Gamma^-(u)$ depends on the directions we assigned to the edges coming out of u in (4.25). Since we do not make use of the labels, not the edge transitions, we shall initialise the parameters for Theorem 3.3.8 in the same way as we did in Section 3.4, meaning our algorithm will work on the same Hilbert space as defined in (3.22):

$$\mathcal{H} = \operatorname{span}\{|u, v\rangle : (u, v) \in E(G')\}. \tag{4.27}$$

Star states: The same is true for our star states, which we will define according to (3.23), meaning that for all $u \in V(G) \setminus \{t\}$ with neighbours (in G') $\Gamma_{G'}(u) = \{v_1, v_2, v_3\}$:

$$|\psi_\star^{G'}(u)\rangle \propto \sqrt{\mathsf{w}_{u,v_1}}(-1)^{\Delta_{u,v_1}}|u, v_1\rangle + \sqrt{\mathsf{w}_{u,v_2}}(-1)^{\Delta_{u,v_2}}|u, v_2\rangle + \sqrt{\mathsf{w}_{u,v_3}}(-1)^{\Delta_{u,v_3}}|u, v_3\rangle. \tag{4.28}$$

For t the star state is similar (as we will show in (4.31)), but slightly different as t only has two neighbours.

The problem is that we cannot efficiently generate all of these star states due to the query access that we have to the graph. For $\ell \in [n-1]_0$ and $v \in V_{2\ell+1} \setminus \{t\} \subset V_{\text{odd}}$ with neighbours $\Gamma(v) = \{u_1, u_2, u_3\}$ there is no immediate cause for alarm yet, as we have, referring to (4.25) and (4.26):

$$|\psi_\star^{G'}(v)\rangle = (-1)^{\ell+1}\frac{1}{\sqrt{3}}\left(|v, u_1\rangle + |v, u_2\rangle + |v, u_3\rangle\right). \tag{4.29}$$

Even though we don't know which layer v is in, nor which neighbour is the parent, we can always generate (4.29) up to a sign difference, which poses no problem if we wish to reflect around its span.

On the other hand, for $\ell \in [n]$ and $u \in V_{2\ell} \setminus \{s\} \subset V_{\text{even}}$, though we can compute $\Gamma_{G'}(u) = \{v_1, v_2, v_3\}$, we do not know which neighbour is the parent – the unique neighbour of u that is further from the centre of the graph than u. Let $p(u) \in \{v_1, v_2, v_3\}$ be the parent of u, and $c_1(u)$, $c_2(u)$ the other two vertices in $\{v_1, v_2, v_3\}$. Then, referring to (4.25), (4.26) and (4.28), the star state of u has the form:

$$|\psi_\star^{G'}(u)\rangle = (-1)^{\ell+1}\sqrt{\frac{2}{3}}\left(|u, p(u)\rangle - \frac{1}{2}|u, c_1(u)\rangle - \frac{1}{2}|u, c_1(u)\rangle\right).$$

Generating this state would require knowing which of the neighbours is the parent, $p(u)$, which is not something that can be learned from simply querying the neighbours of u. However, if we were to weight everything uniformly, our quantum walk would, like a classical random walk, suffer from the fact that the centre of the graph has exponential weight, and most time will be spent there. We avoid this problem by employing the alternative neighbourhoods technique. For $u \in V_{\text{even}} \setminus \{s\}$, define:

$$\forall j \in [3],\ |\psi_\star^j(u)\rangle = \sqrt{\frac{2}{3}}\left(|u, v_j\rangle - \frac{1}{2}|u, v_{j+1}\rangle - \frac{1}{2}|u, v_{j+2}\rangle\right), \tag{4.30}$$

where the indices add modulo 3. Then we know that

$$|\psi_\star^{G'}(u)\rangle \in \{|\psi_\star^1(u)\rangle, |\psi_\star^2(u)\rangle, |\psi_\star^3(u)\rangle\} =: \Psi_\star(u),$$

though we do not know which one.

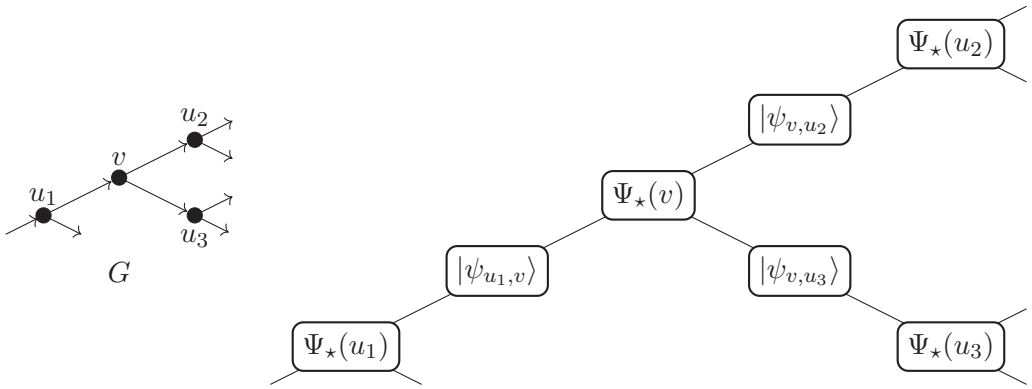


Figure 4.9: A piece of the graph G (left) and the corresponding piece of the overlap graph of the various sets of spaces that form $\Psi_{\mathcal{A}}$ and $\Psi_{\mathcal{B}}$. There is an edge between two nodes in the overlap graph if and only if the sets contain overlapping states. Compare with Figure 4.7.

For s and t, suppose the neighbours (with respect to G') are $\Gamma_{G'}(s) = \{v_1, v_2, s_0\}$ and $\Gamma_{G'}(t) = \{u_1, u_2\}$ – meaning that when we query s, we learn $\{v_1, v_2\}$ and then we add to $\Gamma(s)$ the additional special vertex s_0 such that $v_1 < v_2 < s_0$, and when we query t we learn $\{u_1, u_2\}$. Then the star states are (up to normalisation), respectively:

$$\begin{aligned} |\psi_\star^{G'}(s)\rangle &\propto -\sqrt{\mathsf{w}_0}|s, s_0\rangle + \frac{1}{2}|s, v_1\rangle + \frac{1}{2}|s, v_2\rangle \\ |\psi_\star^{G'}(t)\rangle &\propto -\frac{1}{2}|t, u_1\rangle - \frac{1}{2}|t, u_2\rangle. \end{aligned} \tag{4.31}$$

This is consistent with (4.28) (apart from the fact that t only has two neighbours), since $(s_0, s) \in \overrightarrow{E}(G')$ by definition, and for $i \in \{1, 2\}$, $(s, v_i) \in E_1$, so $(s, v_i) \in \overrightarrow{E}(G)$ since $1 = 1 \mod 4$, and $(u_i, t) \in E_{2n+1}$, so $(u_i, t) \in \overrightarrow{E}(G)$, since $2n + 1 = 1 \mod 4$ (we are assuming n is even), which is why we have minus signs in front of the $|t, u_i\rangle$ (see (4.25)). We can generate $|\psi_\star^{G'}(s)\rangle$ and $|\psi_\star^{G'}(t)\rangle$, because we can recognise s and t. Thus, for all $v \in V_{\text{odd}} \cup \{s\}$ (the vertices with easy to generate star states), we let $\Psi_\star(v) := \{|\psi_\star^{G'}(v)\rangle\}$.

Transition states: Even though we do not make use of labels, nor the transition subroutines from the proof of Theorem 4.3.1, we will define our transition states similarly. Our transition subroutine must apply the transitions $|u, v\rangle \mapsto |v, u\rangle$ for each $\forall (u, v) \in \overrightarrow{E}(G)$. Inspired by looking at the history state construction as in (4.8), but without the extra time register, we define

$$\forall (u, v) \in \overrightarrow{E}(G'),\ |\psi_{u,v}\rangle := |u, v\rangle - |v, u\rangle. \tag{4.32}$$

The star states and the transition states (4.32) will comprise all states of $\Psi^{\mathcal{A}} \cup \Psi^{\mathcal{B}}$ as follows:

$$\begin{aligned} \Psi^{\mathcal{A}} &:= \bigcup_{u \in V(G)} \Psi_\star(u) = \bigcup_{u \in V_{\text{odd}} \cup \{s\}} \{|\psi_\star^{G'}(u)\rangle\} \cup \bigcup_{u \in V_{\text{even}} \setminus \{s\}} \{|\psi_\star^1(u)\rangle, |\psi_\star^2(u)\rangle, |\psi_\star^3(u)\rangle\} \\ \Psi^{\mathcal{B}} &:= \{|\psi_{u,v}\rangle : (u, v) \in \overrightarrow{E}(G')\}. \end{aligned} \tag{4.33}$$

Then $\Psi^{\mathcal{B}}$ is a pairwise orthogonal set, and if we replace each $\Psi_\star(u)$ in $\Psi^{\mathcal{A}}$ with an orthonormal basis for $\text{span}\{\Psi_\star(u)\}$ we get a pairwise orthogonal set. Figure 4.9 shows that the overlap graph for the sets $\Psi_\star(u)$ for $u \in V(G)$ and $\{|\psi_{u,v}\rangle\}$ for $(u, v) \in \overrightarrow{E}(G)$ is bipartite, and we have chosen $\Psi^{\mathcal{A}}$ and $\Psi^{\mathcal{B}}$ according to this bipartition.

Under this choice of $\Psi^{\mathcal{B}}$, we find that $\mathcal{B}$ is precisely the antisymmetric subspace of $\mathcal{H}$, as in Section 3.4. In the remainder of this section, we will show that we can solve the welded trees problem with bounded error by performing phase estimation of $U_{\mathcal{AB}} = (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I)$ on the initial state (chosen as in (3.28))

$$|\psi_0\rangle = \frac{1}{\sqrt{2}}\left(|s_0, s\rangle + |s, s_0\rangle\right),$$

as described in Theorem 3.3.8.

4.4.4 Implementing the unitary

In order to implement $U_{\mathcal{AB}}$, we need to be able to generate an orthonormal basis for each of $\mathcal{A}$ and $\mathcal{B}$, for which we use the following fact.

Claim 4.4.4. *Let $\omega_3 = e^{2\pi i/3}$ be a third root of unity. For a vertex $u \in V(G)$ with neighbours $v_1 < v_2 < v_3$, define for $j \in \{0, 1, 2\}$:*

$$|\widehat{\psi^j}(u)\rangle := \frac{1}{\sqrt{3}}\left(|u, v_1\rangle + \omega_3^j |u, v_2\rangle + \omega_3^{2j} |u, v_3\rangle\right).$$

Then these three vectors are an orthonormal set, and for $u \in V_{\text{even}} \setminus \{s\}$,

$$\text{span}\{|\psi_\star^0(u)\rangle, |\psi_\star^1(u)\rangle, |\psi_\star^2(u)\rangle\} = \text{span}\{|\widehat{\psi^1}(u)\rangle, |\widehat{\psi^2}(u)\rangle\}.$$

For $v \in V_{\text{odd}} \setminus \{t\}$,

$$|\psi_\star^{G'}(v)\rangle \in \text{span}\{|\widehat{\psi^0}(u)\rangle\}.$$

Proof. Note that $\{|\widehat{\psi^0}(u)\rangle, |\widehat{\psi^1}(u)\rangle, |\widehat{\psi^2}(u)\rangle\}$ is an orthonormal basis – it is in fact the Fourier basis (see Definition 2.2.1) – for $\text{span}\{|u, v_1\rangle, |u, v_2\rangle, |u, v_3\rangle\}$. Thus, the first part of the statement is simply proven by observing that for each $j \in \{0, 1, 2\}$, $|\psi_\star^j(u)\rangle \in \text{span}\{|u, v_1\rangle, |u, v_2\rangle, |u, v_3\rangle\}$ and $\langle\psi_\star^j(u)|\widehat{\psi^0}(u)\rangle = 0$; and that $\text{span}\{|\psi_\star^j(u)\rangle\}_{j=0}^2$ has dimension greater than 1. The second statement follows easily from (4.29). □

Lemma 4.4.5. *The unitary $U_{\mathcal{AB}} = (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I)$ can be implemented in $O(1)$ queries to O_G, and $O(n)$ elementary operations.*

Proof. Let

$$\mathcal{H}' = \text{span}\{|j\rangle|u, v\rangle : j \in \{0, 1, 2\}, u \in V(G), v \in \Gamma(u) \cup \{\perp\}\},$$

so in particular $|0\rangle \otimes \mathcal{H} \subset \mathcal{H}'$ (where $\mathcal{H}$ is as in (4.27)). We first show how to implement $2\Pi_{\mathcal{A}} - I$, by describing a unitary $U_\star$ on $\mathcal{H}'$, and in particular, its behaviour on states of the form $|j\rangle|u, \perp\rangle$, where $j = 0$ whenever $u \in V_{\text{odd}} \cup \{s\}$, and $j \in \{1, 2\}$ whenever $u \in V_{\text{even}} \setminus \{s\}$. We begin by querying the neighbours of u in an auxiliary register, Q, initialised to $|0\rangle$ using O_G:

$$|j\rangle|u, \perp\rangle|0\rangle_Q \mapsto |j\rangle|u, \perp\rangle|v_1, v_2, v_3\rangle_Q$$

where if $u \in \{s, t\}$, $v_1 < v_2$ and $v_3 = \perp$ (which we can interpret as s_0 when $u = s$), and otherwise, since we assume $u \in V(G)$, $v_1 < v_2 < v_3$ are the neighbours of u. We initialise an auxiliary register A, and compute a trit $|a\rangle_A$ for $a \in \{0, 1, 2\}$ as follows, to determine what happens next. If $v_3 \neq \perp$, then $a = 0$. Else if $u = 0^{2n} = s$, we let $a = 1$. Else if $v_3 = \perp$ but $u \neq 0^{2n}$, so $u = t$, we let $a = 2$.

Controlled on $|0\rangle_A$, apply QFT_3 (see Definition 2.2.1) to $|j\rangle$ to obtain $|\hat{j}\rangle = (|1\rangle + \omega_3^j|2\rangle + \omega_3^{2j}|3\rangle)/\sqrt{3}$. Then, still conditioned on $|0\rangle_A$, swap the first and third registers, so now the first register contains $|\perp\rangle$, and then perform $|\perp\rangle \mapsto |0\rangle$ on the first register to get

$$|0\rangle|u\rangle|\hat{j}\rangle|0\rangle|v_1, v_2, v_3\rangle_Q|0\rangle_A.$$

Then, conditioned on the value in the $|\hat{j}\rangle$ register, we can copy over the first, second or third value in the $|v_1, v_2, v_3\rangle$ register to get

$$\frac{1}{\sqrt{3}}|0\rangle|u\rangle\left(|1\rangle|v_1\rangle + \omega_3^j|2\rangle|v_2\rangle + \omega_3^{2j}|3\rangle|v_3\rangle\right)|v_1, v_2, v_3\rangle_Q|0\rangle_A.$$

This requires $O(n)$ basic operations. We can uncompute the value $|i\rangle$ in $|i\rangle|v_i\rangle$ by referring to the last register to learn v_i's position, and then we are left with:

$$|0\rangle|\widehat{\psi^j}(u)\rangle|v_1, v_2, v_3\rangle_Q|0\rangle_A.$$

Next, we control on $|1\rangle_A$, meaning $u = s$. In that case, we assume that $j = 0$. Using v_1 and v_2 in the last register, we can map $|\perp\rangle$ to a state proportional to $\sqrt{\mathsf{w}_0}|v_0\rangle + \frac{1}{2}|v_1\rangle + \frac{1}{2}|v_2\rangle$ to get

$$|0\rangle|\psi_\star^{G'}(s)\rangle|v_1, v_2, v_3\rangle_Q|1\rangle_A.$$

Lastly, we control on $|2\rangle_A$, meaning $u = t$. We can compute $g(t)$ in a separate register, and using $g(t)$, v_1, and v_2, map $|\perp\rangle$ to a state proportional to: $-\frac{1}{2}|v_1\rangle - \frac{1}{2}|v_2\rangle$ to get

$$|0\rangle|\psi_\star^{G'}(t)\rangle|v_1, v_2, v_3\rangle_Q|2\rangle_A.$$

We can uncompute the A register, since the registers containing u, and v_1, v_2, v_3 haven't changed. Since the register containing u has not changed, we can uncompute the register $|v_1, v_2, v_3\rangle_Q$ using another call to O_G. Then, we have performed a map, $U_\star$ that acts, for $j = 0$ when $u \in V_{\text{odd}} \cup \{s\}$ and $j \in \{1, 2\}$ when $u \in V_{\text{even}} \setminus \{s\}$, as $|j\rangle|u, \perp\rangle \mapsto |0\rangle|\widehat{\psi^j}(u)\rangle$, where, using Claim 4.4.4, for all $u \in V_{\text{odd}} \cup \{s\}$,

$$\text{span}\{|\widehat{\psi^0}(u)\rangle\} = \text{span}\{|\psi_\star^{G'}(u)\rangle\} = \text{span}\{\Psi_\star(u)\}$$

and for all $u \in V_{\text{even}} \setminus \{s\}$:

$$\text{span}\{|\widehat{\psi^1}(u)\rangle, |\widehat{\psi^2}(u)\rangle\} = \text{span}\{|\psi_\star^1(u)\rangle, |\psi_\star^2(u)\rangle, |\psi_\star^3(u)\rangle\} = \text{span}\{\Psi_\star(u)\}.$$

Thus, $U_\star$ maps the subspace

$$\mathcal{L} := \text{span}\{|0, u, \perp\rangle : u \in (V_{\text{odd}} \cup \{s\}) \setminus M\} \cup \{|1, u, \perp\rangle, |2, u, \perp\rangle : u \in V_{\text{even}} \setminus \{s\}\}$$

of $\mathcal{H}'$ to $|0\rangle \otimes \text{span}\{\Psi^{\mathcal{A}}\} \cong \mathcal{A}$, and thus, $2\Pi_{\mathcal{A}} - I = U_\star\,(2\Pi_{\mathcal{L}} - I)\,U_\star^\dagger$.

We describe how to implement $2\Pi_{\mathcal{L}} - I$. First, initialise four auxiliary flag qubits $|0\rangle_{F_1}|0\rangle_{F_2}|0\rangle_{F_3}|0\rangle_{F_4}$. For a computational basis state $|j\rangle|u, v\rangle$ of $\mathcal{H}'$, by assumption (which is removed at the end of this section) we can efficiently check whether u is in V_{odd} or V_{even}, and we can check whether $u = s = 0^{2n}$ or $u \in M$ in $O(n)$ cost. If $u \in V_{\text{odd}} \cup \{s\} \setminus M$, we check if the first register is 0, and if not, flip F_1 to get $|1\rangle_{F_1}$. If $u \in V_{\text{even}} \setminus \{s\}$, we check if the first register is 1 or 2, and if not, flip F_2 to get $|1\rangle_{F_2}$. If the last register is not $\perp$, flip F_3 to get $|1\rangle_{F_3}$. Lastly if $u \in M \cup \{s_0\}$, flip F_4 to get $|1\rangle_{F_4}$. Reflect if any flag is set, and then uncompute all flags. This can all be done in $O(n)$ basic operations.

Next, we describe how to implement $2\Pi_{\mathcal{B}} - I$, by describing a unitary $U_{\mathcal{S}}$ on $\mathcal{H}'$, and in particular, its behaviour on states of the form $|1\rangle|u, v\rangle$ for $(u, v) \in E(G')$ with $u < v$.

First, apply a Hadamard gate to the first register, and then, controlled on its value, swap the second two registers to get

$$\left(|0\rangle|u,v\rangle - |1\rangle|v,u\rangle\right)/\sqrt{2}.$$

We can uncompute the first register by adding in a bit indicating if the last two registers are in sorted order, to get

$$|0\rangle\frac{1}{\sqrt{2}}\left(|u,v\rangle - |v,u\rangle\right) \in \begin{cases} \operatorname{span}\{|0\rangle|\psi_{u,v}\rangle\} & \text{if } (u,v) \in \overrightarrow{E}(G') \\ \operatorname{span}\{|0\rangle|\psi_{v,u}\rangle\} & \text{if } (v,u) \in \overrightarrow{E}(G'). \end{cases}$$

Thus, $U_{\mathcal{S}}$ maps

$$\mathcal{L}' := \operatorname{span}\{|1\rangle|u,v\rangle : (u,v) \in E(G'), u < v\}$$

to $\operatorname{span}\{|0\rangle|\psi_{u,v}\rangle : (u,v) \in \overrightarrow{E}(G')\} \cong \mathcal{B}$, and so $2\Pi_{\mathcal{B}} - I = U_{\mathcal{S}}\left(2\Pi_{\mathcal{L}'} - I\right)U_{\mathcal{S}}^{\dagger}$. To implement $(2\Pi_{\mathcal{L}'} - I)$, it is enough to check that the first register is 1, and u and v are in sorted order (we know $(u,v) \in E(G')$ by the structure of $\mathcal{H}'$). This can be done in $O(n)$ basic operations. □

4.4.5 Positive analysis

In the case when t is marked, so $M = \{t\} \neq \emptyset$, we exhibit a positive witness (see Definition 3.3.5) $|w\rangle$ that is orthogonal to all states in $\Psi_{\mathcal{A}} \cup \Psi_{\mathcal{B}}$, and that has non-zero overlap with $|\psi_0\rangle = \frac{1}{\sqrt{2}}\left(|s_0,s\rangle + |s,s_0\rangle\right)$. Once again, we construct the positive witness from any s-t flow θ on G, that we extend to an s_0-t flow θ' on G' by sending one unit of flow from s_0 to s. By rescaling the corresponding flow state $|\theta'\rangle$, as defined in (3.30), we obtain our choice of positive witness:

$$|w\rangle = \frac{1}{\sqrt{2\left(\mathcal{E}(\theta) + 1/\mathsf{w}_0\right)}}\left(\sum_{(u,v)\in\overrightarrow{E}(G)} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}\left(|u,v\rangle + |v,u\rangle\right) + \frac{1}{\mathsf{w}_0}\left(|s_0,s\rangle + |s,s_0\rangle\right)\right). \tag{4.34}$$

As seen in (3.32), this state is a potential candidate for our positive witness, as it has non-zero overlap with the initial state $|\psi_0\rangle$:

$$\langle\psi_0|w\rangle = \frac{1}{\sqrt{\mathsf{w}_0\mathcal{E}(\theta) + 1}}. \tag{4.35}$$

Note that in Section 3.4, we chose a rescaled version of $|\theta'\rangle$ to be the positive witness, specifically $\sqrt{\mathsf{w}_0\mathcal{E}(\theta)+1}|\theta'\rangle$. This rescaling was necessary because the definition of the optimal positive witness in Definition 3.3.9 required its inner product with the initial state to equal 1. However, since we are not concerned with the optimal positive witness here, we avoid applying this rescaling. Interestingly, the flow we will construct shortly corresponds to the optimal positive witness, as we will prove in Section 6.3.

Returning to our positive analysis, the more complicated constraint to verify for our positive witness, is that $|w\rangle$ must be orthogonal to both $\mathcal{A}$ and $\mathcal{B}$ (we require orthogonality since we will be constructing a 0-positive witness). By construction, for any s-t flow f on G, our choice of $|w\rangle$ is always orthogonal to $\mathcal{B}$, as it lives in the symmetric subspace of $\mathcal{H}$ (see (3.27)), and as seen in (3.31), it is orthogonal to all star states $|\psi_\star^{G'}(u)\rangle$ for $u \in V(G) \setminus M$. However, due to the technique of alternative neighbourhoods, there are now additional states $|\psi_\star^j(u)\rangle \in \Psi_{\mathcal{A}}$, and in order to be orthogonal to all of these, the flow must satisfy additional constraints. We will show that all these constraints are satisfied by

the natural choice of flow that, for each vertex, comes in from the parent and then sends half to each child. That is, letting E_k for $k \in [2n+1]$ be as in (4.24), and $E_0 = \{(v_0, s)\}$, define

$$\forall k \in [2n+1], (u,v) \in E_k, \quad \theta_{u,v} := \frac{1}{|E_k|} = 2^{-k}. \tag{4.36}$$

Then we first prove the following:

Claim 4.4.6. *Let $u \in V_{\text{even}} \setminus \{s\}$, and let $|w_u\rangle = (|u\rangle\langle u| \otimes I)|w\rangle$. Then $|w_u\rangle \propto |\widehat{\psi}^0(u)\rangle$.*

Proof. Since $u \notin \{s,t\}$, we have

$$|w_u\rangle \propto \sum_{v\in\Gamma^+(u)} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}|u,v\rangle + \sum_{v\in\Gamma^-(u)} \frac{\theta_{v,u}}{\sqrt{\mathsf{w}_{u,v}}}|u,v\rangle = \sum_{v\in\Gamma(u)} (-1)^{\Delta_{u,v}} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}|u,v\rangle.$$

using $\theta_{u,v} = -\theta_{v,u}$, $\mathsf{w}_{u,v} = \mathsf{w}_{v,u}$, and $\Delta_{u,v} = 0$ if $v \in \Gamma^+(u)$, and 1 otherwise. Recall that u has three neighbours: a parent $p(u)$ and two children $c_1(u)$ and $c_2(u)$. Since $u \in V_{2\ell}$ for some ℓ, the edges adjacent to u are (up to direction) in $E_{2\ell}$ and $E_{2\ell+1}$. If ℓ is even, $2\ell = 0 \bmod 4$ and $2\ell+1 = 1 \bmod 4$, so by (4.24), $\Delta_{p(u),u} = \Delta_{u,c_1(u)} = \Delta_{u,c_2(u)} = 0$ if $\ell \in [n/2]$ (i.e. u is in the left tree, so its parent is to its left) and $= 1$ otherwise. If ℓ is odd, $2\ell = 2 \bmod 4$ and $2\ell+1 = 3 \bmod 4$, so $\Delta_{p(u),u} = \Delta_{u,c_1(u)} = \Delta_{u,c_2(u)} = 1$ if $\ell \in [n/2]$ and $= 0$ otherwise. Thus, since $(-1)^{\Delta_{u,p(u)}} = -(-1)^{\Delta_{p(u),u}}$, we always have

$$|w_u\rangle \propto \pm\left(-\frac{\theta_{u,p(u)}}{\sqrt{\mathsf{w}_{u,p(u)}}}|u,p(u)\rangle + \frac{\theta_{u,c_1(u)}}{\sqrt{\mathsf{w}_{u,c_1(u)}}}|u,c_1(u)\rangle + \frac{\theta_{u,c_2(u)}}{\sqrt{w_{u,c_2(u)}}}|u,c_2(u)\rangle\right).$$

Suppose $\ell \in [n/2]$, so u is in the left tree. Then $(p(u),u) \in E_{2\ell}$, so we have

$$\theta_{u,p(u)} = -\theta_{p(u),u} = -\frac{1}{|E_{2\ell}|} = -2^{-2\ell} \text{ and } \sqrt{\mathsf{w}_{u,p(u)}} = \sqrt{\mathsf{w}_{2\ell}} = 2^{-\lceil 2\ell/2\rceil} = 2^{-\ell}$$

by (4.26), and for $i \in \{1,2\}$, $(u,c_i(u)) \in E_{2\ell+1}$, so we have

$$\theta_{u,c_i(u)} = \frac{1}{|E_{2\ell+1}|} = 2^{-(2\ell+1)} \text{ and } \sqrt{\mathsf{w}_{u,c_i(u)}} = \sqrt{w_{2\ell+1}} = 2^{-\lceil (2\ell+1)/2\rceil} = 2^{-(\ell+1)}$$

also by (4.26). Thus:

$$\begin{aligned}|w_u\rangle &\propto \pm\left(-\frac{-2^{-2\ell}}{2^{-\ell}}|u,p(u)\rangle + \frac{2^{-(2\ell+1)}}{2^{-(\ell+1)}}|u,c_1(u)\rangle + \frac{2^{-(2\ell+1)}}{2^{-(\ell+1)}}|u,c_2(u)\rangle\right)\\ &= \pm 2^{-\ell}\left(|u,p(u)\rangle + |u,c_1(u)\rangle + |u,c_2(u)\rangle\right).\end{aligned}$$

On the other hand, if $\ell \in \{n/2+1,\dots,n\}$, so that u is in the right tree, we have $(u,p(u)) \in E_{2\ell+1}$, so:

$$\theta_{u,p(u)} = \frac{1}{|E_{2\ell+1}|} = 2^{-(2n+2-2\ell-1)} \text{ and } \sqrt{\mathsf{w}_{u,p(u)}} = \sqrt{\mathsf{w}_{2\ell+1}} = 2^{-(n+1-\ell)},$$

and for $i \in \{1,2\}$, $(c_i(u),u) \in E_{2\ell}$, so:

$$\theta_{u,c_i(u)} = -\theta_{c_i(u),u} = -\frac{1}{|E_{2\ell}|} = -2^{-(2n+2-2\ell)}$$

and

$$\sqrt{\mathsf{w}_{u,c_i(u)}} = \sqrt{\mathsf{w}_{2\ell}} = 2^{-(n+2-\lceil \frac{2\ell}{2}\rceil)} = 2^{-(n+2-\ell)}.$$

Thus:

$$|w_u\rangle \propto \pm\left(-\frac{2^{-(2n+1-2\ell)}}{2^{-(n+1-\ell)}}|u,p(u)\rangle + \frac{-2^{-(2n+2-2\ell)}}{2^{-(n+2-\ell)}}|u,c_1(u)\rangle + \frac{-2^{-(2n+2-2\ell)}}{2^{-(n+2-\ell)}}|u,c_2(u)\rangle\right)$$
$$= \mp 2^{n-\ell}\left(|u,p(u)\rangle + |u,c_1(u)\rangle + |u,c_2(u)\rangle\right).$$

Hence, letting $\{v_1,v_2,v_3\} = \{p(u),c_1(u),c_2(u)\}$ with $v_1 < v_2 < v_3$, for any $\ell \in [n]$, if $u \in V_{2\ell}$, we have $|w_u\rangle \propto |u,v_1\rangle + |u,v_2\rangle + |u,v_3\rangle$. □

Then we have the following:

Lemma 4.4.7. *Suppose $M = \{t\}$, and let $|w\rangle$ be as defined in (4.34) with respect to the flow defined in (4.36). Then $|w\rangle$ is a 0-positive witness (see Definition 3.3.5) with:*

$$\frac{\||w\rangle\|^2}{|\langle w|\psi_0\rangle|^2} = O(\mathsf{w}_0 n) + 1.$$

Proof. To show that $|w\rangle$ is a 0-positive witness, we must show that it is orthogonal to all states in $\Psi_{\mathcal{A}} \cup \Psi_{\mathcal{B}}$. For $(u,v) \in \overrightarrow{E}(G)$, it is clear from the definition of $|w\rangle$, and the definition of $|\psi_{u,v}\rangle = |u,v\rangle - |v,u\rangle$ (see (4.32)) that $\langle w|\psi_{u,v}\rangle = 0$, meaning $|w\rangle \in \mathcal{B}^{\perp}$.

Recall that we already know from (3.31), that $\langle \psi_\star^{G'}(u)|w\rangle = 0$ for all $u \in V(G)\setminus\{t\}$, if θ is an s_0-t flow on G'. So we now simply check that θ, as defined, is indeed an s_0-t flow on G'. Suppose $u \in V_k$ for some $k \in [n]$. Then u has three neighbours: a parent $p(u) \in V_{k-1}$, and two children $c_1(u), c_2(u) \in V_{k+1}$. We have

$$\theta_{u,p(u)} = -\theta_{p(u),u} = -\frac{1}{|E_k|}, \text{ and } \theta_{u,c_1(u)} = \theta_{u,c_2(u)} = \frac{1}{|E_{k+1}|} = \frac{1}{2|E_k|},$$

and thus

$$\theta_{u,p(u)} + \theta_{u,c_1(u)} + \theta_{u,c_2(u)} = 0.$$

The case for $k \in \{n+1,\dots,2n\}$ is nearly identical, meaning θ is conserved at all $u \in V(G)\setminus\{t\}$. Lastly, θ is indeed a unit flow with its source at s and its sink at t, since

$$\theta_{s_0} = \theta_{s_0,s} = 1, \qquad \theta_t = \theta_{t,c_1(t)} + \theta_{t,c_2(t)} = -\frac{1}{2} - \frac{1}{2} = -1.$$

It thus only remains to show orthogonality of $|w\rangle$ with the states of $\Psi_{\mathcal{A}}$ that are not star states of G'. The only such states are those in (4.30) (some of which are also star states of G'). By Claim 4.4.4, it is sufficient to show orthogonality with the states $|\widehat{\psi^j}(u)\rangle$, for $j \in \{1,2\}$ and $u \in V_{\text{even}}\setminus\{s\}$. Then letting $v_1 < v_2 < v_3$ be the neighbours of u, and appealing to Claim 4.4.6:

$$\sqrt{3}\langle w|\psi_\star^i(u)\rangle = \langle w_u|\left(|u,v_1\rangle + \omega_3^j|u,v_2\rangle + \omega_3^{2j}|u,v_3\rangle\right)$$
$$\propto \left(\langle u,v_1| + \langle u,v_2| + \langle u,v_3|\right)\left(|u,v_1\rangle + \omega_3^j|u,v_2\rangle + \omega_3^{2j}|u,v_3\rangle\right)$$
$$\propto 1 + \omega_3^j + \omega_3^{2j} = 0.$$

Since by (4.35) we know that $\langle\psi_0|w\rangle = 1/\sqrt{\mathsf{w}_0\mathcal{E}(\theta)+1}$ and flow states, and hence also $|w\rangle$, are by construction normalised, we only need to compute $\mathcal{E}(\theta)$ to complete the analysis of its complexity. Here we make use of the fact that all edges in E_k have the same weight, w_k (see (4.26)), and flow, $\frac{1}{|E_k|}$:

$$\mathcal{E}(\theta) = \sum_{(u,v)\in\overrightarrow{E}(G)} \frac{\theta_{u,v}^2}{\mathsf{w}_{u,v}} = \sum_{k=1}^{2n+1} |E_k|\frac{1}{|E_k|^2}\frac{1}{\mathsf{w}_k}$$
$$= \sum_{k=1}^{n}\frac{1}{2^k}\frac{1}{2^{-2\lceil k/2\rceil}} + \sum_{k=n+1}^{2n+1}\frac{1}{2^{2n+2-k}}\frac{1}{2^{-2(n+2-\lceil k/2\rceil)}} = O(n).$$ □

Remark 4.4.8. *The reader may wonder why the weights change by a factor of 4 every two layers, rather than by a factor of 2 every layer. If we set all the weights to 1, the positive witness size is constant, while the negative witness size is exponential. If we change weights by a factor of two at each layer, the negative witness size is constant, whereas the positive witness size is exponential. With the setting of weights that we have chosen, both witness sizes are linear in n (up to scaling by w_0). This setting of weights and edge directions creates a perfect duality between positive and negative witnesses. For vertices $u \in V_{\mathrm{odd}}$, we include the star state, which is proportional to $|\widehat{\psi}^0(u)\rangle$ (see Claim 4.4.4) in $\Psi_{\mathcal{A}}$, so the flow through u must be in $\mathrm{span}\{|\widehat{\psi}^1(u)\rangle, |\widehat{\psi}^2(u)\rangle\}$. Conversely, for vertices $u \in V_{\mathrm{even}}$, we include $\mathrm{span}\{|\widehat{\psi}^1(u)\rangle, |\widehat{\psi}^2(u)\rangle\}$ in $\Psi_{\mathcal{A}}$, so the flow through u must be proportional to $|\widehat{\psi}^0(u)\rangle$.*

4.4.6 Negative analysis

For the negative analysis, it would be sufficient to upper bound the total weight of G and appeal to Theorem 4.3.1, but we will instead explicitly construct a negative witness (see Definition 3.3.2) in order to appeal to Theorem 3.3.8. We choose our negative witness according to (3.39), meaning:

$$|w_{\mathcal{A}}\rangle := -\sum_{u \in V(G)} \frac{\sqrt{2\mathsf{w}_u^{G'}}}{\sqrt{\mathsf{w}_0}} |\psi_\star^{G'}(u)\rangle,$$

$$|w_{\mathcal{B}}\rangle := \frac{1}{\sqrt{2}} \left(|s_0, s\rangle - |s, s_0\rangle\right) + \sum_{(u,v) \in \vec{E}(G)} \frac{\sqrt{2\mathsf{w}_{u,v}}}{\sqrt{\mathsf{w}_0}} \left(|u, v\rangle - |v, u\rangle\right).$$

Then we prove the following:

Lemma 4.4.9. *Suppose $M = \emptyset$. Then $|w_{\mathcal{A}}\rangle, |w_{\mathcal{B}}\rangle$ form a 0-negative witness with $\||w_{\mathcal{A}}\rangle\|^2 = O(n/\mathsf{w}_0) + 2$.*

Proof. When $M = \emptyset$ (that is, $t \notin M$), the star state $|\psi_\star^{G'}(t)\rangle$ lies in $\mathcal{A}$. As a result, each $|\psi_\star^{G'}(u)\rangle$ lies in $\mathcal{A}$ for $u \in V(G)$, meaning so does $|w_{\mathcal{A}}\rangle$. By construction it is also straightforward to see that $|w_{\mathcal{B}}\rangle$ lives in $\mathcal{B}$, i.e. the antisymmetric subspace of $\mathcal{H}$. We know, referring to (3.40), that for this choice of $|w_{\mathcal{A}}\rangle$ and $|w_{\mathcal{B}}\rangle$, we have

$$|w_{\mathcal{A}}\rangle + |w_{\mathcal{B}}\rangle = \sqrt{2}|s, s_0\rangle + \frac{1}{\sqrt{2}} \left(|s_0, s\rangle - |s, s_0\rangle\right) = \frac{1}{\sqrt{2}} \left(|s_0, s\rangle + |s, s_0\rangle\right) = |\psi_0\rangle.$$

We can analyse the complexity of this witness by computing an upper bound on $\||w_{\mathcal{A}}\rangle\|^2$:

$$\begin{aligned}
\||w_{\mathcal{A}}\rangle\|^2 &= \frac{2}{\mathsf{w}_0} \left\| -\sqrt{\mathsf{w}_0}|s, s_0\rangle + \sum_{(u,v) \in E(G)} (-1)^{\Delta_{u,v}} \sqrt{\mathsf{w}_{u,v}} |u, v\rangle \right\|^2 \\
&= \frac{2}{\mathsf{w}_0} \sum_{e \in E(G)} \mathsf{w}_e + 2 = \frac{2}{\mathsf{w}_0} \sum_{k=0}^{2n+1} |E_k| \mathsf{w}_k + 2 \\
&= \frac{2}{\mathsf{w}_0} \sum_{k=0}^{n} \frac{2^k}{2^{2\lceil k/2 \rceil}} + \frac{4}{w_0} \sum_{k=n+1}^{2n+1} \frac{2^{2n+1-k+1}}{2^{2n+4-2\lceil k/2 \rceil}} + 2 = O(n/\mathsf{w}_0) + 2
\end{aligned}$$

using the fact that edges in E_k have weight w_k defined in (4.26), and $|E_k|$ in (4.24). □

4.4.7 Conclusion of the proof

We now apply Theorem 3.3.8 to conclude the proof of Theorem 4.4.2. By Lemma 4.4.7, there is some constant c such that setting $\mathsf{w}_0 = 1/(cn)$, whenever $M = \{t\}$, there exists a positive witness $|w\rangle$ with

$$\frac{\||w\rangle\|^2}{|\langle w|\psi_0\rangle|^2} \leq c_+ := 2.$$

Then by Lemma 4.4.9, there is some

$$\mathcal{C}_- = O(n/\mathsf{w}_0) + 2 = O(n^2)$$

such that whenever $M = \emptyset$, there exists a negative witness with $\||w_{\mathcal{A}}\rangle\|^2 \leq \mathcal{C}_-$. Since the initial state $|\psi_0\rangle = \frac{1}{\sqrt{2}}(|s_0, s\rangle + |s, s_0\rangle)$ can be prepared in $\mathsf{S}_q = 0$ queries and $\mathsf{S} = O(n)$ time, and by Lemma 4.4.5, the unitary can be implemented in $O(1)$ query to O_G, and $O(n)$ time, the phase estimation algorithm distinguishes between the cases $M = \emptyset$ and $M = \{t\}$ in

$$O\left(0 + \sqrt{\mathcal{C}_-}\right) = O(n) \text{ and } O\left(n + \sqrt{\mathcal{C}_-}n\right) = O(n^2)$$

queries and time respectively.

4.4.8 Removing the Assumption that $u \in V_{\text{even}}$ can be Checked

We do not actually require an extra assumption that the algorithm can efficiently check, for a vertex u, if it is in V_{even} or V_{odd}. Intuitively, this is because if a walker starts at u, she can always keep track of the parity of the distance from u, by keeping track of a bit that is initially 0, and flips every time she takes a step. More precisely, we can define a graph G_0 as follows:

$$V(G_0) = V_{\text{even}} \times \{0\} \cup V_{\text{odd}} \times \{1\}$$

$$E(G_0) = \{\{(u, 0), (v, 1)\} : (u, v) \in E(G), u \in V_{\text{even}}\},$$

so that a walk on G_0 is like a walk on G, except that there is a bit indicating which of the two independent sets we are in, which we flip at every step. To find the neighbours of any vertex (u, b), simply query O_G and append $b \oplus 1$ to each of the three returned strings. We let $(s, 0)$ and $(t, 1)$, which are both in $V(G_0)$, take the places of s and t.

CHAPTER 5

Application: k-distinctness

God schiep de dag, en wij sleepten ons erdoorheen.

Dimitri Verhulst, De helaasheid der dingen

This chapter is based on Section 5 in the paper *Multidimensional Quantum Walks, with Application to k-Distinctness* [JZ23], which is joint work with Stacey Jeffery.

The quantum query complexity of k-distinctness is known to be $O(n^{\frac{3}{4}-\frac{1}{4}\frac{1}{2^k-1}})$ for any constant $k \geq 4$. However, this upper bound was established existentially rather than constructively, leaving the efficient implementation of the required unitaries unclear. Consequently, it was not possible to analyse the time cost of this algorithm. The best previously known upper bound on the time complexity of k-distinctness was $\widetilde{O}(n^{1-1/k})$. In this chapter, we present a new upper bound of $\widetilde{O}(n^{\frac{3}{4}-\frac{1}{4}\frac{1}{2^k-1}})$ on the time complexity, matching the best-known query complexity upper bound up to polylogarithmic factors. This improvement is achieved by designing a quantum walk algorithm using our novel multidimensional quantum walk framework introduced in the previous chapter. Since both the construction and analysis of our algorithm are technically involved, we first address the 3-distinctness case as a useful warm-up.

5.1 Introduction to the problem

In the problem of *element distinctness*, the input is a list of n integers, and the output is a bit indicating whether the integers are all distinct, or there exists a pair of integers that are the same, called a *collision*. This problem has been studied as a fundamental problem in query complexity, but also for its relationship to other more practical problems, such as sorting, or *collision finding*, which is similar, but one generally assumes there are many collisions and one wants to find one. In the worst case, element distinctness requires $\Theta(n)$ classical queries [Ajt05].

The first quantum algorithm that managed to improve upon this was a $O(n^{3/4})$ query algorithm [BDH+05], which is a variation of an optimal quantum algorithm for collision finding [BHT97], whose main technique is amplitude amplification [BHMT02]. The algorithm of [BDH+05] could also be implemented time efficiently, in $\widetilde{O}(n^{3/4})$ steps, with a log factor overhead from storing large subsets of the input in a sorted data structure. This was later improved to $O(n^{2/3})$ queries, and $\widetilde{O}(n^{2/3})$ time by Ambainis [Amb07], which is optimal [AS04]. Ambainis' algorithm has been modified to solve other problems in various domains, from k-sum [CE05], to path finding in isogeny graphs [Tan09, CLN16]. Moreover, this algorithm was a critical step in our understanding of quantum query complexity, and quantum algorithms in general, as the algorithm used a new technique that was later generalised by Szegedy into a generic speedup for random walk search algorithms of a particular form [Sze04].

For any constant integer $k \geq 2$, the problem of *k-distinctness* is to decide if an input list of integers contains k copies of the same integer. More formally, given an input $x \in [q]^n$, for some $q \in \mathsf{poly}(n)$, the problem is to decide whether there exist distinct $a_1, \ldots, a_k \in [n]$ such that $x_{a_1} = \cdots = x_{a_k}$, called a k-collision. A search version of this problem asks that the algorithm find a k-collision if one exists. When $k = 2$, this is exactly element distinctness. The search and decision versions are equivalent up to log factors, so we focus on the decision version.

Ambainis [Amb07] actually gave a quantum algorithm for k-distinctness for any $k \geq 2$, with query complexity $O(n^{1-1/(k+1)})$, and time complexity $\widetilde{O}(n^{1-1/(k+1)})$. For $k \geq 3$, Belovs gave an improved quantum query upper bound of $O(n^{3/4-\frac{1}{4}\frac{1}{2^k-1}})$ [Bel12a], however, this upper bound was not constructive. Belovs proved this upper bound by exhibiting a dual adversary solution, which can be turned into a quantum algorithm that relies on controlled calls to a particular unitary. This unitary can be implemented in one query, but actually implementing this algorithm requires giving an efficient circuit for the unitary, which is not possible in general. This is analogous to being given a classical table of values, but no efficient circuit description. While it seems reasonable to guess that the time complexity of k-distinctness should not be significantly higher than the query complexity – what could one possibly do aside from querying and subsequently sorting well-chosen sets of inputs? – the problem of finding a matching time upper bound was open for ten years.

In the meantime, the hardness of the problem was also studied. Lower bounds of $\Omega(n^{\frac{3}{4}-\frac{1}{2k}})$ for $k \geq 3$ [BKT18] and $\Omega(n^{\frac{3}{4}-\frac{1}{4k}})$ for $k \geq 4$ [MTZ20] were exhibited. Progress was also made for the $k = 3$ case specifically when it comes to time complexity. Two simultaneous works, [Bel13] and [CJKM13] (published together as [BCJ+13]), gave a $\widetilde{O}(n^{5/7})$ time upper bound for 3-distinctness. Ref. [Bel13] achieved this bound using the electric network framework, which we discussed in Section 3.4. Ref. [CJKM13] used the MNRS quantum walk framework [MNRS11], and could also be generalised to give a slight improvement on the time upper bound to $\widetilde{O}(n^{1-1/k})$ for any $k > 3$ [Jef14]. These state-of-the-art upper and lower bounds on the query and time complexity, up to this work, are visualised in Figure 5.1.

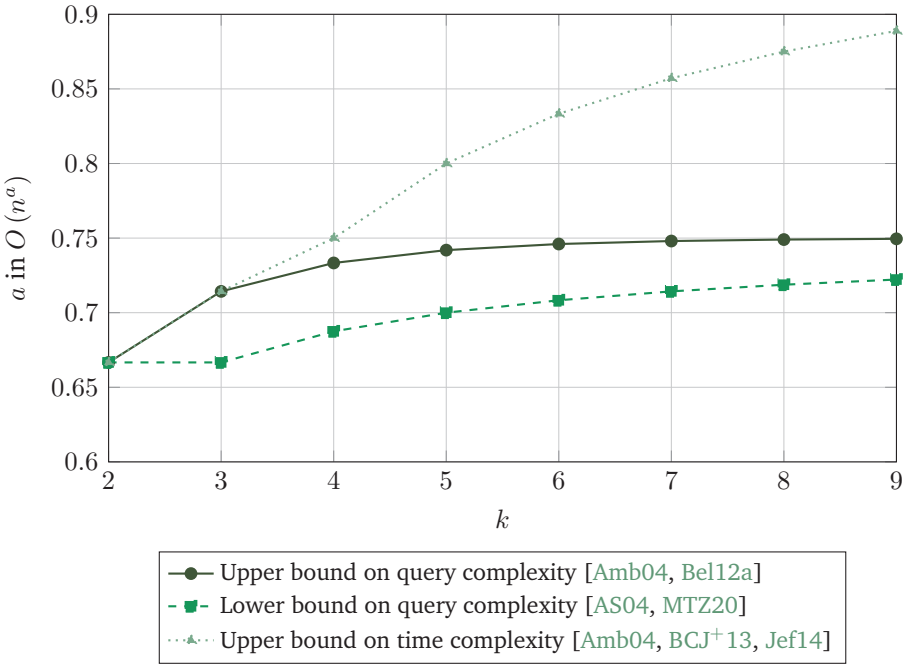


Figure 5.1: State-of-the-art upper and lower bounds on query and time complexity, up to this work. The upper bound on time complexity ignores polylogarithmic factors in n.

In this chapter, we give an upper bound of $\widetilde{O}(n^{\frac{3}{4}-\frac{1}{4}\frac{1}{2^k-1}})$ on the time complexity of k-distinctness (see Theorem 5.5.1), matching the best known query upper bound up to polylogarithmic factors. We do this by applying the framework of multidimensional quantum walks from Chapter 4, specifically Theorem 4.3.1. As a warm-up, we describe the $k = 3$ case of our algorithm in Section 5.4, before giving the full algorithm in Section 5.5. First, we describe some assumptions on the structure of the input in Section 5.3.

5.2 Probability theory

The transition subroutines that we will employ in our quantum walk for k-distinctness, and even 3-distinctness, will not be deterministic and will hence run with some errors, as discussed in Section 4.3.1. To deal with the analysis of these errors, we make use of the following facts from probability theory.

Hypergeometric distribution: In the hypergeometric distribution, specified by the parameters (N, K, d), we draw d objects uniformly without replacement from a set of N objects, K of which are marked, and consider the number of marked objects that are drawn. We will use the following:

Lemma 5.2.1 (Hypergeometric Tail Bounds [JLR11])**.** *Let Z be a hypergeometric random variable with parameters (N, K, d), and $\mu = \frac{Kd}{N}$. Then for every $B \geq 7\mu$, $\Pr[Z \geq B] \leq e^{-B}$. Furthermore, for every $\epsilon > 0$,*

$$\Pr[|Z - \mu| \geq \epsilon\mu] \leq 2\exp\{-((1+\epsilon)\log(1+\epsilon) - \epsilon)\mu\}.$$

We will make use of the second bound from Lemma 5.2.1 in the following form, when $\mu = o(1)$:

Corollary 5.2.2. *Let Z be a hypergeometric random variable with parameters (N, K, d), and $\mu = \frac{Kd}{N}$. Then $\Pr[Z \geq c] \leq 2e^c(c/\mu)^{-c}$.*

d-wise independence It will be convenient to divide the input into blocks, which we will argue are random. In order to avoid the $\widetilde{\Theta}(n)$ cost of sampling a uniform random permutation to define these blocks, we use a d-wise independent family of permutations.

Definition 5.2.3. *Let $\{\tau_s : [n] \to [n]\}_{s \in \mathcal{S}}$ for some finite seed set $\mathcal{S}$. For $d \in \mathbb{N}$ and $\delta \in (0,1)$, we say that τ_s is a d-wise δ-independent permutation (family) if for s chosen uniformly at random from $\mathcal{S}$, for any distinct $i_1, \ldots, i_d \in [n]$ and distinct $i'_1, \ldots, i'_d \in [n]$,*

$$\left| \Pr[\tau_s(i_1) = i'_1, \ldots, \tau_s(i_d) = i'_d] - \frac{1}{n(n-1)\ldots(n-d+1)} \right| \leq \delta.$$

For $d \in \mathsf{polylog}(n)$, and any $\delta \in (0,1)$, there exist families of d-wise δ-independent permutations with the following properties (see, for example, [KNR09]):

- s can be sampled in $O(d \log n \log \frac{1}{\delta})$ time and space.
- For any $s \in \mathcal{S}$, $i \in [n]$, $\tau_s(i)$ can be computed in time $\mathsf{poly}(d \log n \log \frac{1}{\delta})$.
- For any $s \in \mathcal{S}$, $i' \in [n]$, $\tau_s^{-1}(i')$ can be computed in time $\mathsf{poly}(d \log n \log \frac{1}{\delta})$.

For an example of the third property, in d-wise independent permutation families based on Feistel networks applied to d-wise independent functions h, inverting τ_h is as easy as computing h. We will design our algorithms assuming such a construction for $\delta = 0$, although this is not known to exist. By taking δ to be a sufficiently small inverse polynomial, our algorithm will not notice the difference.

5.3 Assumptions on the input

To simplify our algorithm (slightly unsuccessfully that is, as it still occupies an entire chapter in this thesis), we assume that either there is no k-collision, or there is a unique k-collision, $a_1, \ldots, a_k \in [n]$. This is justified by the following lemma, which follows from [Amb07, Section 5].

Lemma 5.3.1. *Fix constants $k \geq 2$ and $\lambda \in [1/2, 1)$. Let $\mathcal{A}$ be an algorithm that decides k-distinctness in bounded error with complexity $\widetilde{O}(n^\lambda)$ when there is at most one k-collision. Then there is an algorithm $\mathcal{A}'$ that decides k-distinctness (in the general case) in bounded error in complexity $\widetilde{O}(n^\lambda)$.*

This fact has been exploited in nearly every quantum algorithm for k-distinctness. Another standard trick is to assume that $[n]$ is partitioned as

$$[n] = A_1 \cup \cdots \cup A_k$$

such that the unique k-collision $(a_1, \ldots, a_k)$, (if it exists) is in $A_1 \times \cdots \times A_k$. Towards fixing **Problem 1** from Section 4.2.3, we further partition each of $A_2, \ldots, A_{k-1}$ as

$$A_\ell = A_\ell^{(1)} \cup \cdots \cup A_\ell^{(m_\ell)}$$

for some m_ℓ. We will choose these partitions as follows. Fix a d-wise independent permutation $\tau : [n] \to [n]$, for $d = \log^{2^{k-1}}(n)$ that is both efficiently computable, and efficiently invertible (see Definition 5.2.3 and the discussion below). For $\ell \in [k]$, define:

$$A_\ell = \{\tau(i) : i \in \{(\ell-1)n/k + 1, \ldots, \ell n/k\}\}$$

and for $j \in [m_\ell]$, define:

$$A_\ell^{(j)} = \left\{ \tau(i) : i \in \left\{ (\ell-1)n/k + (j-1)\frac{n}{km_\ell} + 1, \ldots, (\ell-1)n/k + j\frac{n}{km_\ell} \right\} \right\}. \tag{5.1}$$

We make use of the following facts about these partitions:

Lemma 5.3.2. *1. For any $i \in [n]$, we can check to which $A_\ell^{(j)}$ it belongs in $\mathsf{polylog}(n)$ complexity.*

2. For any $\ell \in [k]$, we can generate a uniform superposition over A_ℓ, and for any $j \in [m_\ell]$, we can generate a uniform superposition over $A_\ell^{(j)}$, in $\mathsf{polylog}(n)$ complexity.

3. $\Pr[a_1 \in A_1, \ldots, a_k \in A_k] = \Omega(1)$.

Proof. Since $d \in \mathsf{polylog}(n)$, we can assume (see discussion below Definition 5.2.3) that both τ and τ^{-1} can be computed in $\mathsf{polylog}(n)$ complexity. Then for Item 1, it is enough to compute $\tau^{-1}(i)$.

For Item 2, we describe how to perform a superposition over $\{\tau(i) : i \in \{\ell, \ldots, r\}\}$ for any integers $\ell < r$. First generate the uniform superposition over the set $\{\ell, \ldots, r\}$, and compute τ in a new register, to get (up to normalisation) $\sum_{i=\ell}^{r} |i\rangle|\tau(i)\rangle$. Then uncompute the first register by computing τ^{-1} of the second register and adding it into the first.

Finally, Item 3 follows from the $d > k$-wise independence of τ. □

For any disjoint subsets of $[n]$, $S_1, \ldots, S_\ell$, define:

$$\mathcal{K}(S_1, \ldots, S_\ell) = \{(i_1, \ldots, i_\ell) \in S_1 \times \cdots \times S_\ell : x_{i_1} = \cdots = x_{i_\ell}\}. \tag{5.2}$$

Then without loss of generality, we can assume that for each $A_j^{(\ell)}$, $\mathcal{K}(A_1, \ldots, A_{j-1}, A_j^{(\ell)}) = \Theta(|A_j^{(\ell)}|)$, because we can simply pad the input with $\Theta(n)$ extra $(k-1)$-collisions, evenly spread across the blocks.

5.4 Warm-up: 3-distinctness algorithm

In this section, we prove an upper bound on the time complexity of 3-distinctness.

Theorem 5.4.1. *There is a quantum algorithm that decides 3-distinctness with bounded error in $\widetilde{O}(n^{5/7})$ complexity.*

This upper bound is not new, having been proven in [BCJ+13], but its proof in our new framework is a useful warm-up for Section 5.5, where we generalise the algorithm to all constants $k > 3$. Throughout this section, $\widetilde{O}$ will suppress polylogarithmic factors in n. Our algorithm will roughly follow the one described in Section 4.2.3, but with the modifications, also briefly mentioned in Section 4.2.3, needed to circumvent the problems with the approach, for which we need the multidimensional quantum walk framework, Theorem 4.3.1. We start by setting up these modifications, before formally defining the graph that will be the basis for our quantum walk algorithm, and then performing the necessary analysis to apply Theorem 4.3.1.

Recall from Section 4.2.3 that the basic idea of our quantum walk algorithm is to walk on sets $R = (R_1, R_2)$ where $R_1 \subset A_1$ and $R_2 \subset A_2$.

Towards fixing Problem 1: The first problem identified in Section 4.2.3 is that $|R_2|$ is larger than the total time we would like our algorithm to spend, meaning we do not want to spend $|R_2|$ steps sampling and writing down the set R_2. To this end, we have partitioned A_2 into equal sized blocks:

$$A_2 = A_2^{(1)} \cup \cdots \cup A_2^{(m_2)},$$

(see Section 5.3 for details of how this partition is chosen). We redefine R_2 as follows: whenever we want to choose a subset of A_2, we do so by selecting $R_2 \subset [m_2]$, which encodes the subset of A_2:

$$\overline{R}_2 := \bigcup_{j \in R_2} A_2^{(j)}.$$

We choose m_2 so that $|A_2^{(j)}| = \frac{n}{3m_2}$ is large enough so that for a random set R_1 of size r_1, the expected size of $\mathcal{K}(R_1, A_2^{(j)})$ is constant, so we set $m_2 = \Theta(r_1)$. Finally, we choose $t_2 = |R_2|$ so that $|\overline{R}_2| = t_2\frac{n}{3m_2}$ is the desired size of R_2 (denoted r_2 in [Bel12a]) and for consistency also define $t_1 = r_1$. We will find that the optimal parameter settings are $t_1 = n^{5/7}$ and $t_2 = n^{4/7}$ (so $m_2 = \Theta(n^{5/7})$).

Towards fixing Problem 2: In order to solve the second problem discussed in Section 4.2.3, following a similar construction in [Bel12a], each of R_1 and R_2 will be a *tuple* of disjoint sets, as follows. We have $R_1 = (R_1(\{1\}), R_1(\{2\}), R_1(\{1,2\}))$ where $R_1(\{1\})$, $R_1(\{2\})$, and $R_1(\{1,2\})$ are disjoint subsets of A_1 of size t_1; and $R_2 = (R_2(1), R_2(2))$, where $R_2(1)$ and $R_2(2)$ are disjoint subsets of $[m_2]$ of size t_2 (note that this alters $|R_1|$ and $|R_2|$ by a constant factor), meaning for $s \in \{1,2\}$,

$$\overline{R}_2(s) := \bigcup_{j \in R_2(s)} A_2^{(j)}$$

are disjoint subsets of A_2 of size $t_2\frac{n}{3m_2}$. We also use R_1 and R_2 to denote the union of sets in the tuple, so for example, $j \in \overline{R}_2$ means $j \in \overline{R}_2(1) \cup \overline{R}_2(2)$.

For a vertex labelled by $R = (R_1, R_2)$, we maintain *data* with the following components. We query everything in R_1, so for $S \in 2^{\{1,2\}} \setminus \emptyset$, we define:

$$\begin{aligned} D_1(R_1(S)) &:= \{(i_1, x_{i_1}) : i_1 \in R_1(S)\} \\ D_1(R) &:= (D_1(R_1(\{1\})), D_1(R_1(\{2\})), D_1(R_1(\{1,2\}))) \end{aligned}$$

and for $s \in \{1,2\}$ define

$$\begin{aligned} D_2(R_2(s)|R_1) &:= \bigcup_{S \subseteq \{1,2\} : s \in S} \{(i_1, i_2, x_{i_1}) : i_2 \in \overline{R}_2(s), i_1 \in R_1(S), x_{i_1} = x_{i_2}\} \\ D_2(R) &:= (D_2(R_2(1)|R_1), D_2(R_2(2)|R_1)). \end{aligned} \tag{5.3}$$

Finally we let

$$D(R) := (D_1(R), D_2(R)).$$

So to summarise, we query everything in R_1, but we only query those things in $\overline{R}_2$ that have a collision in R_1, and even then, not in every case: if $i_2 \in \overline{R}_2(s)$, we only query it if it has a collision with $R_1(\{s\})$ or $R_1(\{1,2\})$ (see Figure 4.5). This partially solves **Problem 2**, because it ensures that if we choose to add a new index i_1 to R_1, we have three choices of where to add it, and either all of those choices are fine (they don't introduce a *fault* in $D_2(R)$), or exactly one of them is fine.

For a finite set $\mathcal{S}$, and positive integers r and ℓ, we will use the notation

$$\binom{\mathcal{S}}{r^{(\ell)}} := \binom{\mathcal{S}}{\underbrace{r, \ldots, r}_{\ell \text{ times}}} \tag{5.4}$$

to denote the set of all ℓ-tuples of disjoint subsets of $\mathcal{S}$, each of size r. Finally, we define:

$$\binom{\mathcal{S}}{r^{(\ell)}}^{+} := \bigcup_{\ell'=1}^{\ell} \left(\binom{\mathcal{S}}{r^{(\ell'-1)}, r+1, r^{(\ell-\ell')}} \right), \tag{5.5}$$

to be the set of all ℓ-tuples of disjoint sets of $\mathcal{S}$ such that exactly one of the sets has size $r+1$, and all others have size r. We let $\mu(S)$ denote the smallest element of S.

5.4.1 The graph G

We now define G, by defining disjoint vertex sets $V_0, V_0^+, V_1, V_2, V_3$ whose union will make up $V(G)$, as well as the edges between adjacent sets.

V_0: We first define

$$V_0 := \left\{ v^0_{R_1,R_2} = (0, R_1, R_2, D(R_1,R_2)) : (R_1,R_2) \in \binom{A_1}{t_1^{(3)}} \times \binom{[m_2]}{t_2^{(2)}} \right\} \tag{5.6}$$

on which the initial distribution will be uniform: $\sigma(v^0_{R_1,R_2}) = \frac{1}{|V_1|}$. We implicitly store all sets including R_1, R_2 and $D(R_1,R_2)$ in a data structure with the properties described in Section 3.5.1. This will only be important when we analyze the time complexity of the setup and transition subroutines.

V_0^+ and $E_0^+ \subset V_0 \times V_0^+$: Next, each vertex in V_0^+ will be labeled by a vertex in V_0, along with an index $i_1 \notin R_1$ that we have decided to add to one of $R_1(\{1\})$, $R_1(\{1,2\})$ or $R_1(\{2\})$. We have not yet decided to which of the three sets it will be added (nor added it):

$$V_0^+ := \left\{ v^0_{R_1,R_2,i_1} := ((0,+), R_1, R_2, D(R_1,R_2), i_1) : v^0_{R_1,R_2} \in V_0, i_1 \in A_1 \setminus R_1 \right\},$$
$$\text{so } |V_0^+| = |V_0|(n/3 - 3t_1). \tag{5.7}$$

There is an edge between $v^0_R \in V_0$ and $v^0_{R,i_1} \in V_0^+$ for any $i_1 \in A_1 \setminus R_1$, and for any $v^0_{R,i_1} \in V_0^+$, $v^0_R \in V_0$ is its unique in-neighbour, so we define edge label sets (see Definition 3.2.13)

$$L^+(v^0_R) := A_1 \setminus R_1 \quad \text{and} \quad L^-(v^0_{R,i_1}) := \{\leftarrow\},$$

and let $f_{v^0_R}(i_1) = v^0_{R,i_1}$, and $f_{v^0_{R,i_1}}(\leftarrow) = v^0_R$. We let E_0^+ be the set of all such edges,

$$E_0^+ := \left\{ (v^0_R, v^0_{R,i_1}) : v^0_R \in V_0, i_1 \in A_1 \setminus R_1 \right\},$$

and set $\mathsf{w}_e = \mathsf{w}_0^+ = 1$ for all $e \in E_0^+$. This together with (5.7) implies that

$$|E_0^+| = |V_0^+| = |V_0|(n/3 - 3t_1). \tag{5.8}$$

V_1 and $E_1 \subset V_0^+ \times V_1$: Continuing, vertices in V_1 represent having added an additional index to R_1, so we define:

$$V_1(S) := \left\{ v^1_{R_1,R_2} = (1, R_1, R_2, D(R_1,R_2)) : \right. \tag{5.9}$$
$$\left. (R_1,R_2) \in \binom{A_1}{t_1^{(3)}}^+ \times \binom{[m_2]}{t_2^{(2)}}, |R_1(S)| = t_1 + 1 \right\},$$
$$V_1 := \bigcup_{S \in 2^{\{1,2\}} \setminus \{\emptyset\}} V_1(S)$$
$$\text{so } |V_1| = 3\binom{n/3}{t_1+1, t_1, t_1}\binom{m_2}{t_2,t_2} = 3\frac{n/3 - 3t_1}{t_1+1}\binom{n/3}{t_1,t_1,t_1}\binom{m_2}{t_2,t_2} = \frac{n-9t_1}{t_1+1}|V_0|. \tag{5.10}$$

For a vertex $v^0_{R,i_1} \in V_0^+$ we have chosen an index i_1 to add to R_1, but we have not yet decided to which part of R_1 it should be added. A transition to a vertex in V_1 consists of choosing an $S \in 2^{\{1,2\}} \setminus \{\emptyset\}$ and adding i_1 to $R_1(S)$, so

$$L^+(v^0_{R,i_1}) := 2^{\{1,2\}} \setminus \{\emptyset\},$$

and $f_{v^0_{R,i_1}}(S) = v^1_{R'}$, where R' is obtained from R by inserting i_1 into $R_1(S)$. Note that not all of these labels represent edges with non-zero weight, as we want to ensure that adding i_1 to $R_1(S)$ does not introduce a *fault*, meaning that adding i_1 to $R_1(S)$ should not require that any collision involving i_1 be added to $D_2(R)$.

Viewing transitions in E_1 from the other direction, a vertex $v^1_{R'} \in V_1(S)$ is connected to a vertex $v^0_{R,i_1} \in V_0^+$ if we can obtain R from R' by removing i_1 from $R'_1(S)$, and if doing so does not require an update to $D_2(R')$, meaning there do not exist any $s \in S$ and $i_2 \in \overline{R}'_2(s)$ such that $x_{i_1} = x_{i_2}$. So for any $v^1_{R'} \in V_1(S)$, we let

$$\begin{aligned} L^-(v^1_{R'}) :&= \{i_1 \in R'_1(S) : \nexists s \in S, i_2 \in \overline{R}'_2(s) \text{ s.t. } x_{i_1} = x_{i_2}\} \\ &= \{i_1 \in R'_1(S) : \nexists i_2 \text{ s.t. } (i_1, i_2, x_{i_1}) \in D_2(R')\}, \end{aligned} \tag{5.11}$$

and $f_{v^1_{R'}}(i_1) = v^0_{R'\setminus\{i_1\},i_1}$. It is currently not clear how to define E_1, the set of (non-zero weight) edges between V_0^+ and V_1, because $|V_0^+| \cdot |L^+(v^0_{R,i_1})| > |V_1| \cdot |L^-(v^1_{R'})|$, so in particular, we cannot assign nonzero weights $\mathsf{w}_{u,i}$ to all $u \in V_0^+$, $i \in L^+(u)$, because that would make E_1 larger than we have labels L^- for. We will instead assign non-zero weights $\mathsf{w}_{v,j}$ to those edges where $v \in V_1$ and $j \in L^-(v)$. That is, define:

$$E_1 := \left\{ \left(f_{v^1_{R'}}(i_1), v^1_{R'} \right) = \left(v^0_{R'\setminus\{i_1\},i_1}, v^1_{R'} \right) : v^1_{R'} \in V_1, i_1 \in L^-(v^1_{R'}) \right\}$$

and give weight $\mathsf{w}_e = \mathsf{w}_1 = 1$ to all $e \in E_1$. This means that for $u = v^0_{R,i_1} \in V_0^+$, there are some $S \in 2^{\{1,2\}} \setminus \{\emptyset\}$ with $\mathsf{w}_{u,S} = 0$ – namely those with $f_v^{-1}(u) \notin L^-(v)$ for $v = f_u(S)$. To investigate which S this applies to, we introduce for all $v^0_{R,i_1} \in V_0^+$:

$$\mathcal{I}(v^0_{R,i_1}) := \left\{ s \in \{1,2\} : \exists i_2 \in \overline{R}_2(s) \text{ s.t. } x_{i_2} = x_{i_1} \right\}, \tag{5.12}$$

so $\mathcal{I}(v^0_{R,i_1})$ consists of those $s \in \{1,2\}$ where a fault occurs if i_1 is added to $R_1(S)$ such that $s \in S$. Note that since we assume the unique 3-collision has a part in A_3, i_1 can have at most one colliding element in $\overline{R}_2$, and so it cannot be in both $\overline{R}_2(1)$ and $\overline{R}_2(2)$, which are disjoint. Thus, $\mathcal{I}(v^0_{R,i_1}) \subsetneq \{1,2\}$ – so it is $\emptyset$, $\{1\}$, or $\{2\}$ (this heavy-handed notation is overkill here, but we are warming up for k-distinctness, where it is necessary). We now have the following:

Lemma 5.4.2. *Let $R^{S\leftarrow i_1}$ be obtained from R by inserting i_1 into $R_1(S)$. Then*

$$E_1 = \left\{ \left(v^0_{R,i_1}, v^1_{R^{S\leftarrow i_1}} \right) : v^0_{R,i_1} \in V_0^+, S \in 2^{\{1,2\}\setminus\mathcal{I}(v^0_{R,i_1})} \setminus \{\emptyset\} \right\}.$$

So for all $v^0_{R,i_1} \in V_0^+$, and $S \in L^+(v^0_{R,i_1})$, $\mathsf{w}_{v^0_{R,i_1},S} = \begin{cases} \mathsf{w}_1 = 1 & \text{if } S \cap \mathcal{I}(v^0_{R,i_1}) = \emptyset \\ 0 & \text{else.} \end{cases}$

Proof. Let E'_1 be the right-hand side of the identity in the theorem statement, so we want to show $E_1 = E'_1$. Fix any $v^0_{R,i_1} \in V_0^+$ and $S \in 2^{\{1,2\}\setminus\mathcal{I}(v^0_{R,i_1})} \setminus \{\emptyset\}$, and let $R' = R^{S\leftarrow i_1}$. Then since $S \cap \mathcal{I}(v_{R,i_1}) = \emptyset$, by definition of $\mathcal{I}(v_{R,i_1})$ there does not exist any $s \in S$ and $i_2 \in \overline{R}'_2(s)$ such that $x_{i_1} = x_{i_2}$. Hence, $L^-(v^1_{R'})$, which implies $E'_1 \subseteq E_1$.

For the other direction, fix any $v^1_{R'} \in V_1(S)$ and $i_1 \in L^-(v^1_{R'})$. Since $i_1 \in R'_1(S)$, we have $v^0_{R'\setminus\{i_1\},i_1} \in V_0^+$ and $(R' \setminus \{i_1\})^{S\leftarrow i_1} = R'$. Since by definition of $L^-(v^1_{R'})$ there does not exist $s \in S$ and $i_2 \in \overline{R}'_2(s)$ such that $x_{i_1} = x_{i_2}$, we immediately have $S \cap \mathcal{I}(v_{R'\setminus\{i_1\},i_1}) = \emptyset$. This implies $E_1 \subseteq E'_1$. □

From (5.7) we now have

$$|E_1| \leq 3|V_0^+| = 3|V_0|(n/3 - 3t_1). \tag{5.13}$$

V_2 and $E_2 \subset V_1 \times V_2$: Vertices $v_R^1 \in V_1(S)$ represent having added an additional index i_1 to $R_1(S)$, so $|R_1(S)| = t_1 + 1$. A vertex $v^2_{R_1,R_2'} \in V_2$ is adjacent to v_R^1 if R_2' is obtained from R_2 by adding $j_2 \notin R_2$ to $R_2(s)$ for some choice of $s \in \{1,2\}$. We will not let this choice of s be arbitrary though and instead, in order to simplify things in the more complicated k-distinctness setting, we require that j_2 be added to $R_2(\mu(S))$, where $\mu(S)$ denotes the minimum element of S.

$$V_2(S) := \Bigg\{ v_R^2 = (2, R, D(R)) : R \in \binom{A_1}{t_1^{(3)}}^+ \times \binom{[m_2]}{t_2^{(2)}}^+, \\ |R_1(S)| = t_1 + 1, |R_2(\mu(S))| = t_2 + 1 \Bigg\},$$

$$V_2 := \bigcup_{S \in 2^{\{1,2\}} \setminus \{\emptyset\}} V_2(S).$$

This means that

$$\begin{aligned} |V_2| &= 3\binom{n/3}{t_1+1, t_1, t_1}\binom{m_2}{t_2+1, t_2} \\ &= 3\frac{n/3 - 3t_1}{t_1+1}\binom{n/3}{t_1,t_1,t_1}\frac{m_2 - 2t_2}{t_2+1}\binom{m_2}{t_2,t_2} = O\left(\frac{nm_2}{t_1t_2}|V_0|\right). \end{aligned} \tag{5.14}$$

We move from $v_R^1 \in V_1$ to $v_{R'}^2 \in V_2$ by selecting some $j_2 \in [m_2] \setminus R_2$ to add to R_2; and from $v_{R'}^2$ to v_R^1 by selecting some j_2 to remove from R_2, so for $v_R^1 \in V_1(S)$ and $v_{R'}^2 \in V_2(S)$, we let

$$L^+(v_R^1) := [m_2] \setminus R_2 \text{ and } L^-(v_{R'}^2) := R_2'(\mu(S)).$$

The sets $L^+(v_R^1)$ and $L^-(v_R^1)$ (defined in (5.11)) should be disjoint, but this does not appear to be the case. To ensure this, we implicitly append a label $\leftarrow$ to every label in $L^-(u)$ for any u, and $\rightarrow$ to every label in $L^+(u)$. We let $f_{v_R^1}(j_2) = v^2_{R_1,R_2^{\mu(S)\leftarrow j_2}}$ when $v_R^1 \in V_1(S)$, and $f_{v^2_{R_1,R_2'}}(j_2) = v^1_{R_1,R_2'\setminus\{j_2\}}$. Accordingly we define $E_2(S)$ to be the set of all such edges:

$$\begin{aligned} E_2 :&= \bigcup_{S \in 2^{\{1,2\}} \setminus \{\emptyset\}} \{(v_R^1, v^2_{R_1,R_2^{\mu(S)\leftarrow j_2}}) : v_R^1 \in V_1(S), j_2 \in [m_2] \setminus R_2\} \\ &= \bigcup_{S \in 2^{\{1,2\}} \setminus \{\emptyset\}} \left\{ \left(v^1_{R_1,R_2'\setminus\{j_2\}}, v^2_{R_1,R_2'} \right) : v^2_{R_1,R_2'} \in V_2(S), j_2 \in R_2(\mu(S)) \right\}. \end{aligned}$$

We set $\mathsf{w}_e = \mathsf{w}_2 = \sqrt{n/m_2}$ for all $e \in E_2$, and observe, using (5.10), that:

$$|E_2| = (m_2 - 2t_2)|V_1| = \frac{(m_2 - 2t_2)(n - 9t_1)}{t_1 + 1}|V_0|. \tag{5.15}$$

The final stage: V_3 and E_3: The last stage is very simple, as every vertex in V_3 represents having added an additional index to each of R_1, R_2 and chosen some $i_3 \in A_3$:

$$V_3 := \{v^3_{R_1,R_2,i_3} = (3, R_1, R_2, D(R_1,R_2), i_3) : v^2_{R_1,R_2} \in V_2, i_3 \in A_3\}.$$

There is an edge between $v_R^2 \in V_2$ and $v^3_{R,i_3} \in V_3$ for any $i_3 \in A_3$, and for any $v^3_{R,i_3} \in V_3$, v_R^2 is its unique (in-)neighbour, so we define

$$L^+(v_R^2) := A_3 \text{ and } L(v^3_{R,i_3}) = L^-(v^3_{R,i_3}) := \{\leftarrow\},$$

u	$j \in L^-(u)$	$f_u(j)$	$i \in L^+(u)$	$f_u(i)$
$v_R^0 \in V_0$	$\emptyset$		$i_1 \in A_1 \setminus R_1$	v_{R,i_1}^0
$v_{R,i_1}^0 \in V_0^+$	$\leftarrow$	v_R^0	$S \in 2^{\{1,2\}} \setminus \{\emptyset\}$	$v_{R^{S\leftarrow i_1}}^1$
$v_R^1 \in V_1(S)$	$i_1 \in R_1(S) : d_R^{\rightarrow}(i_1) = 0$	$v_{R\setminus\{i_1\},i_1}^0$	$j_2 \in [m_2] \setminus R_2$	$v_{R^{\mu(S)\leftarrow j_2}}^2$
$v_R^2 \in V_2(S)$	$j_2 \in R_2(\mu(S))$	$v_{R\setminus\{j_2\}}^1$	$i_3 \in A_3$	v_{R,i_3}^3
$v_{R,i_3}^3 \in V_3$	$\leftarrow$	v_R^{k-1}	$\emptyset$	

Table 5.1: A summary of the vertex sets and the labels of edges coming into and out of each vertex. Foreshadowing Section 5.5, we here define $d_R^{\rightarrow}(i_1)$ to be 0 if and only if there is no $i_2 \in \bigcup_{s\in S} R_2(s)$ such that $x_{i_1} = x_{i_2}$. Here $R^{\mu(S)\leftarrow j_2}$ is obtained from R by inserting j_2 into $R_2(\mu(S))$, where $\mu(S)$ is the minimum element of S. We remark that $L^-(u)$ and $L^+(u)$ should always be disjoint. To ensure that this holds, we implicitly append a $\leftarrow$ label to all of $L^-(u)$ and a $\rightarrow$ label to all of $L^+(u)$.

and let $f_{v_R^2}(i_3) = v_{R,i_3}^3$, and $f_{v_{R,i_3}^3}(\leftarrow) = v_R^2$. We let E_3 be the set of all such edges,

$$E_3 = \left\{ \left(v_R^2, v_{R,i_3}^3\right) : v_R^2 \in V_2, i_3 \in A_3 \right\},$$

and set $\mathsf{w}_e = \mathsf{w}_3 = 1$ for all $e \in E_3$. Then using (5.14) we observe

$$|E_3| = \frac{n}{3}|V_2| = O\left(\frac{n^2 m_2}{t_1 t_2}|V_0|\right). \tag{5.16}$$

The graph G: The full graph G is defined by

$$\begin{aligned} V(G) &= V_0 \cup V_0^+ \cup V_1 \cup V_2 \cup V_3 \\ \text{and } \overrightarrow{E}(G) &= \{(u,v) : u \in V(G), i \in L^+(u), \mathsf{w}_{u,i} \neq 0\} = E_0^+ \cup E_1 \cup E_2 \cup E_3, \end{aligned}$$

where the sets $L^+(u)$ are summarised in Table 5.1, and the condition under which the weight $\mathsf{w}_{u,i} = 0$ can be found in Lemma 5.4.2. Non-zero edge weights are summarised in Table 5.2.

The marked set and checking cost: In the notation of Theorem 4.3.1, we let $V_{\mathsf{M}} = V_3$, and we will define a subset $M \subseteq V_3$ as follows. If $(a_1, a_2, a_3) \in A_1 \times A_2 \times A_3$ is the unique 3-collision (see Section 5.3), we let

$$M = \left\{ v_{R_1,R_2,i_3}^3 \in V_3 : \exists S \in 2^{\{1,2\}} \setminus \{\emptyset\}, \text{ s.t. } a_1 \in R_1(S), a_2 \in \overline{R}_2(\mu(S)), a_3 = i_3 \right\}, \tag{5.17}$$

and otherwise $M = \emptyset$. Recall that $v_{R_1,R_2,i_3}^3 = (3, R_1, R_2, D(R_1,R_2), i_3)$, where $D(R_1,R_2)$ includes $D_2(R)$, defined in (5.3), storing all pairs (i_1, i_2, x_{i_1}) such that $x_{i_1} = x_{i_2}$ and $\exists S \in 2^{\{1,2\}}$ and $s \in S$ with $i_1 \in R_1(S)$ and $i_2 \in R_2(s)$. Thus, we can decide if $v_{R_1,R_2,i_3}^3 \in V_3$ is marked by querying i_3 to obtain x_{i_3} and looking it up (see Section 3.5.1) in $D_2(R)$ to see if we find some (i_1, i_2, x_{i_3}), in which case, it must be that $a_1 = i_1$, $a_2 = i_2$ and $a_3 = i_3$. Thus, the checking cost is at most

$$\mathsf{C} = O(\log n). \tag{5.18}$$

Edge set	Weights	Complexity
$E_0^+ \subset V_0 \times V_0^+$	$\mathsf{w}_0^+ = 1$	$\mathsf{T}_0^+ = \widetilde{O}(1)$
$E_1 \subset V_0^+ \times V_1$	$\mathsf{w}_1 = 1$	$\mathsf{T}_1 = \widetilde{O}(1)$
$E_2 \subset V_1 \times V_2$	$\mathsf{w}_2 = \sqrt{n/m_2}$	$\mathsf{T}_2 = \widetilde{O}(\sqrt{n/m_2})$
$E_3 \subset V_2 \times V_3$	$\mathsf{w}_3 = 1$	$\mathsf{T}_3 = \widetilde{O}(1)$

Table 5.2: A summary of the weights and complexities (see Section 5.4.3) of each edge set.

5.4.2 The star states and their generation

We define a set of alternative neighbourhoods for G (see Definition 4.2.1). For all $u \in V(G) \setminus V_0^+$, we define $\Psi_\star(u) = \{|\psi_\star^G(u)\rangle\}$, which by Table 5.1 is equal to the following: for $u = v^0_{R_1,R_2} \in V_0$,

$$|\psi_\star^G(u)\rangle = \sum_{i_1 \in A_1 \setminus R_1} \sqrt{\mathsf{w}_0^+}|v^0_{R_1,R_2}, i_1\rangle; \tag{5.19}$$

for $u = v^1_{R_1,R_2} \in V_1(S)$,

$$|\psi_\star^G(u)\rangle = - \sum_{\substack{i_1 \in R_1(S): \\ \nexists i_2,\, (i_1,i_2,x_{i_1}) \in D_2(R)}} \sqrt{\mathsf{w}_1}|v^1_{R_1,R_2}, \leftarrow, i_1\rangle + \sum_{j_2 \in [m_2] \setminus R_2} \sqrt{\mathsf{w}_2}|v^1_{R_1,R_2}, \rightarrow, j_2\rangle; \tag{5.20}$$

for $u = v^2_{R_1,R_2} \in V_2(S)$,

$$|\psi_\star^G(u)\rangle = - \sum_{j_2 \in R_2(\mu(S))} \sqrt{\mathsf{w}_2}|v^2_{R_1,R_2}, \leftarrow, j_2\rangle + \sum_{i_3 \in A_3} \sqrt{\mathsf{w}_3}|v^2_{R_1,R_2}, \rightarrow, i_3\rangle; \tag{5.21}$$

and finally for $u = v^3_{R_1,R_2,i_3} \in V_3$,

$$|\psi_\star^G(u)\rangle = -\sqrt{\mathsf{w}_3}|v^3_{R_1,R_2,i_3}, \leftarrow\rangle. \tag{5.22}$$

Here we have explicitly included the $\rightarrow$ and $\leftarrow$ parts of each element of $L^+(u)$ and $L^-(u)$, which are normally left implict, in order to stress that the first and second sum are orthogonal.

From Table 5.1, as well as the description of w from Lemma 5.4.2, we can see that for $u = v^0_{R,i_1} \in V_0^+$,

$$|\psi_\star^G(u)\rangle = -\sqrt{\mathsf{w}_0^+}|u, \leftarrow\rangle + \sum_{S_1 \subseteq \{1,2\} \setminus \mathcal{I}(u): S_1 \neq \emptyset} \sqrt{\mathsf{w}_1}|u, S_1\rangle.$$

To generate this state, one would have to compute $\mathcal{I}(u)$ (see (5.12)), which would require finding any $i_2 \in R_2$ such that $x_{i_1} = x_{i_2}$, which is too expensive. Hence, we simply add all three options, for possibilities $\mathcal{I}(u) \in \{\emptyset, \{1\}, \{2\}\}$ (see also Figure 4.6), to $\Psi_\star(u)$:

$$\begin{aligned} \Psi_\star(u) := \{|\psi_\star^{\emptyset}(u)\rangle &:= \sqrt{\mathsf{w}_0^+}|u, \leftarrow\rangle + \sqrt{\mathsf{w}_1}|u, \{1\}\rangle + \sqrt{\mathsf{w}_1}|u, \{1,2\}\rangle + \sqrt{\mathsf{w}_1}|u, \{2\}\rangle, \\ |\psi_\star^{\{1\}}(u)\rangle &:= \sqrt{\mathsf{w}_0^+}|u, \leftarrow\rangle + \sqrt{\mathsf{w}_1}|u, \{2\}\rangle, \\ |\psi_\star^{\{2\}}(u)\rangle &:= \sqrt{\mathsf{w}_0^+}|u, \leftarrow\rangle + \sqrt{\mathsf{w}_1}|u, \{1\}\rangle\} \ni |\psi_\star^G(u)\rangle. \end{aligned} \tag{5.23}$$

Note that it is important that each state in $\bigcup_{u\in V_0^+}\Psi_\star(u)$ (and therefore each $|\psi_\star^G(u)\rangle$) have at least one outgoing (i.e. forward) edge. Otherwise, it would be impossible to satisfy **P2** of Theorem 4.3.1 (or equivalently, Item 2 of Lemma 5.4.12). This is satisfied because $\mathcal{I}(u)$ is always a proper subset of $\{1,2\}$.

We now describe how to generate the states in $\bigcup_{u\in V(G)}\Psi_\star(u)$ in $\widetilde{O}(1)=\mathsf{polylog}(n)$ complexity (see Definition 4.2.1). We will make use of the following lemma.

Lemma 5.4.3. *Let $V'\subseteq V(G)\setminus V_0\cup V_{\mathsf{M}}$ be such that there exists some constant c such that for all $u\in V'$, $L(u)\subseteq\{0,1\}^c$. Suppose for all $u\in V'$,*

$$\Psi_\star(u)=\{|u\rangle|\phi_\ell\rangle:\ell\in[d']\}$$

for some constant d', and states $|\phi_\ell\rangle\in\mathrm{span}\{|j\rangle:j\in\{0,1\}^c\}$. Then for some $d\le d'$, there are orthonormal bases $\overline{\Psi}(u)=\{|\overline{\psi}_{u,1}\rangle,\dots,|\overline{\psi}_{u,d}\rangle\}$ for $\Psi_\star(u)$, for each $u\in V'$, and a map $U'_\star$ that can be implemented in cost $O(1)$ such that for all $u\in V'$ and $\ell\in[d]$, $U'_\star|u,\ell\rangle=|\overline{\psi}_{u,\ell}\rangle$.

Proof. First note that by the assumptions we are making, $d:=\dim\mathrm{span}\{\Psi_\star(u)\}$ for all $u\in V'$, and d is a constant. Fix any orthonormal basis $\{|\overline{\phi}_1\rangle,\dots,|\overline{\phi}_d\rangle\}$ for $\mathrm{span}\{|\phi_\ell\rangle:\ell\in[d']\}$, which is independent of u. Since the basis lives in a constant-dimensional subspace, the map: $C_\star:|\ell\rangle\mapsto|\overline{\phi}_\ell\rangle$ acts on a constant number of qubits, and so can be implemented in $O(1)$ elementary gates. We complete the proof by letting $U'_\star=I\otimes C_\star$, and observe that: $U'_\star|u,\ell\rangle=|u\rangle|\overline{\phi}_\ell\rangle=:|\overline{\psi}_{u,\ell}\rangle$. □

Lemma 5.4.4. *The states $\Psi_\star=\{\Psi_\star(u)\}_{u\in V(G)}$ can be generated in $\widetilde{O}(1)$ complexity.*

Proof. The description of a vertex $u\in V(G)$ begins with a label indicating to which of V_0,V_0^+,V_1,V_2,V_3 it belongs. Thus, we can define subroutines $U_0,U_{0,+},U_1,U_2,U_3$ that generate the star states in each vertex set respectively, and then $U_\star=\sum_{\ell=0}^3|\ell\rangle\langle\ell|\otimes U_\ell+|0,+\rangle\langle0,+|\otimes U_{0,+}$ will generate the star states in the sense of Definition 4.2.1.

We begin with U_0. For $v_R^0\in V_0$, we have $\Psi_\star(v_R^0)=\{|\psi_\star^G(v_R^0)\rangle\}$, where $|\psi_\star^G(v_R^0)\rangle$ is as in (5.19). Thus, implementing the map $U_0:|v_R^0\rangle|0\rangle\mapsto\propto|\psi_\star^G(v_R^0)\rangle$ is as simple as generating a uniform superposition over A_1, and then using $O(\log n)$ rounds of amplitude amplification to get inverse polynomially close to the uniform superposition over $A_1\setminus R_1$.

For $U_{0,+}$, since all $v_{R,i_1}^0\in V_0^+$ have the same star states, modulo v_{R,i_1}^0 itself, with constant-sized label set $L=\{\{1\},\{2\},\{1,2\},\leftarrow\}$, we can apply Lemma 5.4.3, to get a $U_{0,+}$ that costs $O(1)$.

We continue with U_1. For $v_R^1\in V_1$, we have $\Psi_\star(v_R^1)=\{|\psi_\star^G(v_R^1)\rangle\}$, where $|\psi_\star^G(v_R^1)\rangle$ is as in (5.20). Thus, to implement the map $U_1:|u\rangle|0\rangle\mapsto\propto|\psi_\star^G(u)\rangle$, we first compute (referring to Table 5.2 for the weights):

$$|u,0\rangle\mapsto\propto|u\rangle\left(-\sqrt{\mathsf{w}_1}|\leftarrow\rangle+\sqrt{\mathsf{w}_2}|\rightarrow\rangle\right)|0\rangle=|u\rangle\left(-|\leftarrow\rangle+(n/m_2)^{1/4}|\rightarrow\rangle\right)|0\rangle,$$

which can be implemented by a $O(1)$-qubit rotation. Then conditioned on $\leftarrow$, generate a uniform superposition over $i_1\in R_1$, and then use $O(\log n)$ rounds of amplitude amplification to get inverse polynomially close to a superposition over $i_1\in R_1$ such that there is no $(i_1,i_2,x_{i_1})\in D_2(R)$. We have used the fact that our data structure supports taking a uniform superposition (see Section 3.5.1). Finally, conditioned on $\rightarrow$, generate a uniform superposition over $j_2\in[m_2]\setminus R_2$.

The implementation of U_2 is similar, but instead (see (5.21)) we perform a single qubit rotation to get $-\sqrt{\mathsf{w}_2}|\leftarrow\rangle+\sqrt{\mathsf{w}_3}|\rightarrow\rangle$ in the last register, and then conditioned on the value of this register, we either generate a uniform superposition over $R_2(\mu(S))$ or A_3.

Finally, referring to (5.22), we see that the implementation of U_3 is trivial. We thus conclude that $U_\star$ can be implemented in $\widetilde{O}(1)=\mathsf{polylog}(n)$ complexity. □

5.4.3 The transition subroutines

In this section we show how to implement the transition map $|u,i\rangle \mapsto |v,j\rangle$ for $(u,v) \in \overrightarrow{E}(G)$ with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$ (see Definition 3.2.13). We do this by exhibiting uniform (in the sense of Lemma 3.5.2) subroutines $\mathcal{S}_0^+, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ that implement the transition map for (u,v) in E_0^+, E_1, E_2, E_3 respectively (defined in Section 5.4.1) whose union makes up $\overrightarrow{E}(G)$. In Corollary 5.4.10, we will combine these to get a quantum subroutine (Definition 3.5.1) for the full transition map.

Lemma 5.4.5. *There is a uniform subroutine $\mathcal{S}_0^+$ such that for all $(u,v) \in E_0^+$ with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$, $\mathcal{S}_0^+$ maps $|u,i\rangle$ to $|v,j\rangle$ with error 0 in complexity $\mathsf{T}_{u,v} = \mathsf{T}_0^+ = \widetilde{O}(1)$.*

Proof. For $(v_R^0, v_{R,i_1}^0) \in E_0^+$, $\mathcal{S}_0^+$ should implement the map:

$$\begin{aligned} &|v_R^0, i_1\rangle \mapsto |v_{R,i_1}^0, \leftarrow\rangle \\ \equiv\ &|(0, R, D(R)), i_1\rangle \mapsto |((0,+), R, D(R), i_1), \leftarrow\rangle. \end{aligned}$$

It is easy to see that this can be done in $\mathsf{polylog}(n)$ complexity (and is therefore trivially uniform): we just need to do some accounting to move i_1 from the edge label register to the vertex register, and update the first register $|0\rangle \mapsto |(0,+)\rangle$. □

Lemma 5.4.6. *There is a uniform subroutine $\mathcal{S}_1$ such that for all $(u,v) \in E_1$ with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$, $\mathcal{S}_1$ maps $|u,i\rangle$ to $|v,j\rangle$ with error 0 in complexity $\mathsf{T}_{u,v} = \mathsf{T}_1 = \widetilde{O}(1)$.*

Proof. For $(v_{R_1,R_2,i_1}^0, v_{R_1',R_2}^1) \in E_1$, where $v_{R_1',R_2}^1 \in V_1(S)$, $\mathcal{S}_1$ should implement the map:

$$\begin{aligned} &|v_{R_1,R_2,i_1}^0, S\rangle \mapsto |v_{R_1',R_2}^1, i_1\rangle \\ \equiv\ &|((0,+), R_1, R_2, D(R_1,R_2), i_1), S\rangle \mapsto |(1, R_1, R_2, D(R_1', R_2)), i_1\rangle. \end{aligned}$$

To implement this transition, we need only insert i_1 into $R_1(S)$, query i_1 to obtain x_{i_1} and update the data by inserting (i_1, x_{i_1}) into the $D_1(R)$ part of $D(R_1,R_2) = (D_1(R), D_2(R))$ (see Section 3.5.1). Note that we *do not* attempt to update the $D_2(R)$ part of the data by searching $\overline{R}_2$ for collisions with i_1. If there is some $s \in S$ and $i_2 \in \overline{R}_2(s)$ such that $x_{i_1} = x_{i_2}$, then by definition of E_1, $(v_{R_1,R_2,i_1}^0, v_{R_1',R_2}^1) \notin E_1$. To finish, we uncompute S by checking which of the three parts of R_1 has size $t_1 + 1$, account for the moving of i_1 from the vertex register to the edge label register, and map $|(0,+)\rangle$ to $|1\rangle$ in the first register. The total cost is $\mathsf{polylog}(n)$. □

We now move on to $\mathcal{S}_2$, which is somewhat more complicated. For $(v_{R_1,R_2}^1, v_{R_1,R_2'}^2) \in E_2$, where $v_R^1 \in V_1(S)$, and $R_2'(\mu(S)) = R_2(\mu(S)) \cup \{j_2\}$ for some $j_2 \in [m_2] \setminus R_2$, $\mathcal{S}_2$ should act as

$$\begin{aligned} &|v_{R_1,R_2}^1, j_2\rangle \mapsto |v_{R_1,R_2'}^2, j_2\rangle \\ \equiv\ &|(1, R_1, R_2, D(R_1,R_2)), j_2\rangle \mapsto |(2, R_1, R_2', D(R_1,R_2')), j_2\rangle. \end{aligned} \tag{5.24}$$

The complexity of this map, which we will implement with some error, depends on $|\mathcal{K}(R_1, A_2^{(j_2)})|$ (see (5.2)), the number of collisions to be found between R_1 and the block $A_2^{(j_2)}$, which is implicitly being added to $\overline{R}_2$ by adding j_2 to R_2. Lemma 5.4.7 below describes how to implement this transition map as long as there are fewer than $c_{\max} \log n$ collisions to be found for some constant $c_{\max}$. For the case when $|\mathcal{K}(R_1, A_2^{(j_2)})| \geq c_{\max} \log n$, we will let the algorithm fail (so there is no bound on the error for such transitions). That is, we let:

$$\tilde{E} := \left\{ (v_R^1, v_{R'}^2) \in E_2 : |\mathcal{K}(R_1, A^{(j_2)})| \geq c_{\max} \log n, \text{ where } \{j_2\} = R_2' \setminus R_2 \right\}. \tag{5.25}$$

Lemma 5.4.7. *Fix any constant κ. There is a uniform subroutine $\mathcal{S}_2$ that implements the transition map that maps $|u,i\rangle$ to $|v,j\rangle$ for all $(u,v) \in E_2 \setminus \tilde{E}$, with error $O(n^{-\kappa})$, in complexity $\mathsf{T}_{u,v} = \mathsf{T}_2 = \widetilde{O}(\sqrt{n/m_2})$.*

Proof. To implement the map in (5.24), we need to insert j_2 into $R_2(\mu(S))$ to obtain R_2', update $D_2(R)$ to reflect this insertion, and increment the first register. All of these take $\mathsf{polylog}(n)$ complexity, except for updating $D_2(R)$. To update $D_2(R)$, we need to search $A_2^{(j_2)}$ – the new block we're adding to $\overline{R}_2$ – to find anything that collides with R_1. Since the number of such collisions is less than $c_{\max} \log n$, we can do this using quantum search, which is uniform, with error $O(n^{-\kappa})$ for any desired constant κ in complexity $O(\sqrt{n/m_2} \log^2 n)$, since $|A_2^{(j_2)}| = \sqrt{n/m_2}$. □

Lemma 5.4.8. *For any constant κ, there exists a choice of constant $c_{\max}$ sufficiently large such that $|\tilde{E}| \leq n^{-\kappa}|E_2|$.*

Proof. By Lemma 5.2.1 (or as a special case of Lemma 5.5.6), for every $j_2 \in [m_2]$, if R_1 is uniformly random from $\binom{A_1}{t_1^{(3)}}$, there exists a constant $c_{\max}$ such that $\Pr[|\mathcal{K}(R_1, A_2^{(j_2)})| \geq c_{\max} \log n] \leq n^{-\kappa}$. It follows that the proportion of edges in E_2 that are in $\tilde{E}$ is at most $n^{-\kappa}$. □

Lemma 5.4.9. *There is a subroutine $\mathcal{S}_3$ such that for all $(u,v) \in E_3$ with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$, $\mathcal{S}_3$ maps $|u,i\rangle$ to $|v,j\rangle$ with error 0 in complexity $\mathsf{T}_{u,v} = \widetilde{O}(1)$.*

Proof. The proof is identical to that of Lemma 5.4.5. □

In order to apply Theorem 4.3.1, we need to implement the full transition map as a quantum subroutine in the sense of Definition 3.5.1.

Corollary 5.4.10. *Let κ be any constant. There is a quantum subroutine (in the sense of Definition 3.5.1) that implements the full transition map with errors $\epsilon_e \leq n^{-\kappa}$ for all $e \in \overrightarrow{E}(G) \setminus \tilde{E}$, and times $\mathsf{T}_e = \widetilde{O}(1)$ for all $e \in \overrightarrow{E}(G) \setminus E_2$, and $\mathsf{T}_e = \mathsf{T}_2 = \widetilde{O}(\sqrt{n/m_2})$ for all $e \in E_2$.*

Proof. We combine Lemma 5.4.5, Lemma 5.4.6, Lemma 5.4.7 and Lemma 5.4.9 using Lemma 3.5.3. □

5.4.4 Initial state and setup cost

The state $|\sigma\rangle$, from which the initial state is constructed, is defined to be the uniform superposition over V_0:

$$|\sigma\rangle := \sum_{v^0_{R_1,R_2} \in V_0} \frac{1}{\sqrt{|V_0|}} |v^0_{R_1,R_2}\rangle.$$

Lemma 5.4.11. *The state $|\sigma\rangle$ can be generated with error $n^{-\kappa}$ for any constant κ in complexity*

$$\mathsf{S} = \widetilde{O}\left(t_1 + t_2\sqrt{\tfrac{n}{m_2}}\right).$$

Proof. We start by taking a uniform superposition over all $R_1 \in \binom{A_1}{t_1^{(3)}}$ and $R_2 \in \binom{[m_2]}{t_2^{(2)}}$ stored in data structures as described in Section 3.5.1, and querying everything in R_1 to get $D_1(R)$, which costs $\widetilde{O}(t_1 + t_2)$. Next for each $s \in \{1,2\}$, we search for all elements of $\overline{R}_2(s)$ that collide with an element of $R_1(\{s\})$ or $R_1(\{1,2\})$. However, we do not want to spend too long on this step, so we stop if we find ct_2 collisions, for some constant c. If we do this before all collisions are found, that part of the state is not correct, but we argue

that this only impacts a very small part of the state. The cost of this search is (up to log factors) $\sqrt{t_2|\overline{R}_2|} = t_2\sqrt{n/m_2}$.

For a uniform R_1 and fixed R_2, the expected value of $Z = |\mathcal{K}(R_1, \overline{R}_2)|$, the number of collisions, is

$$\mu = O\left(\frac{|\overline{R}_2|t_1}{n}\right) = O\left(\frac{t_2\frac{n}{m_2}t_1}{n}\right) = O(t_2),$$

since $m_2 = \Theta(t_1)$. Let c' be a constant such that $\mu \le c't_2$, and choose $c = 7c'$. Since Z is a hypergeometric random variable, we have, by Lemma 5.2.1, $\Pr[Z \ge ct_2] \le e^{-ct_2} = o(n^{-\kappa})$ for any κ, since t_2 is polynomial in n. Thus, the state we generate is $n^{-\kappa}$-close to $|\sigma\rangle$. □

5.4.5 Positive analysis

For the positive analysis, we must exhibit a flow (see Definition 3.2.3) from V_0 to M whenever $M \neq \emptyset$.

Lemma 5.4.12. *There exists some $\mathcal{R}^{\mathsf{T}} = \widetilde{O}(|V_0|^{-1})$ such that the following holds. Whenever there is a unique 3-collision $(a_1, a_2, a_3) \in A_1 \times A_2 \times A_3$, there exists a flow θ on G that satisfies conditions **P1**-**P5** of Theorem 4.3.1. Specifically:*

1. *For all $e \in \tilde{E}$, $\theta_e = 0$.*
2. *For all $u \in V(G) \setminus (V_0 \cup M)$ and $|\psi_\star\rangle \in \Psi_\star(u)$, $\langle\psi_\star|\theta\rangle = 0$.*
3. *$\sum_{u\in V_0} \theta_u = 1$.*
4. *$\sum_{u\in V_0} \frac{|\theta_u - \sigma(u)|^2}{\sigma(u)} \le 1$.*
5. *$\mathcal{E}^{\mathsf{T}}(\theta) \le \mathcal{R}^{\mathsf{T}}$.*

Proof. Recall that M is the set of $v^3_{R,i_3} \in V_3$ such that for some $S \subseteq \{1,2\}$, $a_1 \in R_1(S)$, $a_2 \in \overline{R}_2(\mu(S))$ and $a_3 = i_3$. Let $j^* \in [m_2]$ be the unique block label such that $a_2 \in A_2^{(j^*)}$. Then $a_2 \in \overline{R}_2(\mu(S))$ if and only if $j^* \in R_2(\mu(S))$. Assuming $M \neq \emptyset$, we define a flow θ on G with all its sinks in M. It will have sources in both V_1 and M, but all other vertices will conserve flow. This will imply **Item 2** for all *correct* star states of G, $|\psi^G_\star(u)\rangle$, but we will have to take extra care to ensure that **Item 2** is satisfied for the additional states in $\Psi_\star(u) : u \in V_0^+$.

To satisfy condition **P5** of Theorem 4.3.1, we must upper bound $\mathcal{E}^{\mathsf{T}}(\theta) = \mathcal{E}(\theta^{\mathsf{T}})$ (see Definition 3.2.7), which is the energy of the flow θ extended to a graph G^{T}, in which each edge of G in E_2 has been replaced by a path of length $\mathsf{T}_2 = \widetilde{O}(\sqrt{n/m_2})$, and all other edges have been replaced by paths of length $\widetilde{O}(1)$ (see Corollary 5.4.10). We define θ on E_0^+, E_1, E_2 and E_3 stage by stage, and upper bound the contribution to $\mathcal{E}^{\mathsf{T}}(\theta)$ for each stage.

$\mathcal{R}_0^+$, Item 3, and Item 4: Let M_0 be the set of $v^0_{R_1,R_2} \in V_0$ such that $a_1 \notin R_1$, $j^* \notin R_2$, and for $c_{\max}$ as in Lemma 5.4.8, $|\mathcal{K}(R_1, A_1^{(j^*)})| < c_{\max}\log n$ (see (5.2)). This latter condition is because we will later send flow down edges that add j^* to R_2, and we don't want to have flow on edges in $\tilde{E}$. For all $v^0_R \in M_0$, let $\theta_{v^0_R, v^1_{R,a_1}} = |M_0|^{-1}$. For all other edges in E_0^+, let $\theta_e = 0$. Note that we can already see that $\theta_u = |M_0|^{-1}$ for all $u \in M_0$, so we satisfy **Item 3**. By Lemma 5.4.8, we know that the proportion of R_1 that are excluded because $|\mathcal{K}(\overline{R}_1, A_1^{(j^*)})| \ge c_{\max}\log n$ is $o(1)$, so we can conclude:

$$\frac{|V_0|}{|M_0|} = (1+o(1))\left(1+O\left(\frac{t_1}{n}\right)\right)\left(1+O\left(\frac{t_2}{m_2}\right)\right) = 1+o(1). \tag{5.26}$$

Since $\sigma(u) = \frac{1}{|V_0|}$, we can conclude with **Item 4** of the lemma statement:

$$\sum_{u \in V_0} \frac{|\theta_u - \sigma(u)|^2}{\sigma(u)} = |V_0|^2 \left(\frac{1}{|M_0|} - \frac{1}{|V_0|} \right)^2 = \left(\frac{|V_0|}{|M_0|} - 1 \right)^2 = o(1).$$

Using $\mathsf{w}_0^+ = 1$ and $\mathsf{T}_e = \widetilde{O}(1)$ for all $e \in E_0^+$ (see Table 5.2), the contribution of the edges in E_0^+ to the energy of the flow can be computed as

$$\mathcal{R}_0^+ = \sum_{e \in E_0^+} \mathsf{T}_e \frac{\theta_e^2}{\mathsf{w}_0^+} = \widetilde{O}\left(\sum_{u \in M_0} \frac{1}{|M_0|^2} \right) = \widetilde{O}\left(\frac{1}{|M_0|} \right), \tag{5.27}$$

since each vertex in M_0 has a unique outgoing edge with flow, and the flow is uniformly distributed.

$\mathcal{R}_1$ **and Item 2:** Let M_0^+ be the set of $v_{R,i_1}^0 \in V_0^+$ such that $v_R^0 \in M_0$ and $i_1 = a_1$, so $|M_0^+| = |M_0|$. These are the only vertices in V_0^+ that have flow coming in from V_0, and specifically, the incoming flow from V_0 to a vertex in M_0^+ is $\frac{1}{|M_0|}$.

The only way there could be a fault adding a_1 to R_1 would be if $a_2 \in \overline{R}_2$, but we have ensured that that is not the case. Thus, for each $u \in M_0^+$, $\mathcal{I}(u) = \emptyset$, so there are three edges going into V_1 (labelled by $\{1\}$, $\{2\}$, and $\{1,2\}$, all disjoint from $\mathcal{I}(u)$) to which we can assign flow.

Item 2 is satisfied for all $|\psi_\star^G(u)\rangle : u \in V(G) \setminus (V_0 \cup V_3)$ by virtue of θ conserving flow at all vertices in $V(G) \setminus (V_0 \cup V_3)$ (we have not finished defining θ, but it will be defined so that this holds). However, for $u \in V_0^+$, $\Psi_\star(u) = \{|\psi_\star^{\mathcal{I}}(u)\rangle\}_{\mathcal{I} \subsetneq \{1,2\}}$ (see (5.23)) contains more than just $|\psi_\star^G(u)\rangle$. When $u \in V_0^+ \setminus M_0^+$, there is no flow through u, so Item 2 is easily seen to be satisfied for all states in $\Psi_\star(u)$. For $u \in M_0^+$, $|\psi_\star^G(u)\rangle = |\psi_\star^\emptyset(u)\rangle$, so the additional constraints we need to take additional care to satisfy are those for $|\psi_\star^{\{s\}}(u)\rangle$ with $s \in \{1,2\}$:

$$\begin{aligned}
\langle \psi_\star^{\{s\}}(u)|\theta\rangle &\propto \sum_{i \in L^+(u)} \frac{\theta_{u,f_u(i)}}{\sqrt{\mathsf{w}_1}} \langle \psi_\star^{\{s\}}(u)|u,i\rangle - \sum_{j \in L^-(u)} \frac{\theta_{u,f_u(j)}}{\sqrt{\mathsf{w}_0^+}} \langle \psi_\star^{\{s\}}(u)|u,i\rangle && \text{see (4.12)} \\
&= \sum_{S \in 2^{\{1,2\}} \setminus \{\emptyset\}} \frac{\theta_{u,f_u(S)}}{\sqrt{\mathsf{w}_1}} \langle \psi_\star^{\{s\}}(u)|u,S\rangle \sqrt{\mathsf{w}_1} - \frac{\theta_{u,f_u(\leftarrow)}}{\sqrt{\mathsf{w}_0^+}} \langle \psi_\star^{\{s\}}(u)|u,\leftarrow\rangle && \text{see Table 5.1} \\
&= \frac{\theta_{u,f_u(\{3-s\})}}{\sqrt{\mathsf{w}_1}} \sqrt{\mathsf{w}_1} - \frac{\theta_{u,f_u(\leftarrow)}}{\sqrt{\mathsf{w}_0^+}} \left(-\sqrt{\mathsf{w}_0^+} \right) && \text{see (5.23)} \\
&= \theta_{u,v_{\{3-s\}}} + \theta_{u,v^0},
\end{aligned}$$

where $v^0 = f_u(\leftarrow)$ is the neighbour of u in V_0, and $v_{\{3-s\}} = f_u(\{3-s\})$ is the neighbour of u in V_1 with edge labelled by $\{3-s\}$ (see Figure 5.2). So for $s' \in \{1,2\}$, we must have

$$0 = \theta_{u,v_{\{s'\}}} + \theta_{u,v^0} = \theta_{u,v_{\{s'\}}} - \frac{1}{|M_0|},$$

since $\theta_{u,v^0} = -\theta_{v^0,u} = -\frac{1}{|M_0|}$. To satisfy this, we set:

$$\theta_{u,v_{\{1\}}} = \theta_{u,v_{\{2\}}} = \frac{1}{|M_0|},$$

meaning that all the flow that comes into u along edge (v^0,u) must leave u along edge $(u,v_{\{1\}})$, but it must also all leave along edge $(u,v_{\{2\}})$. However, we have now assigned

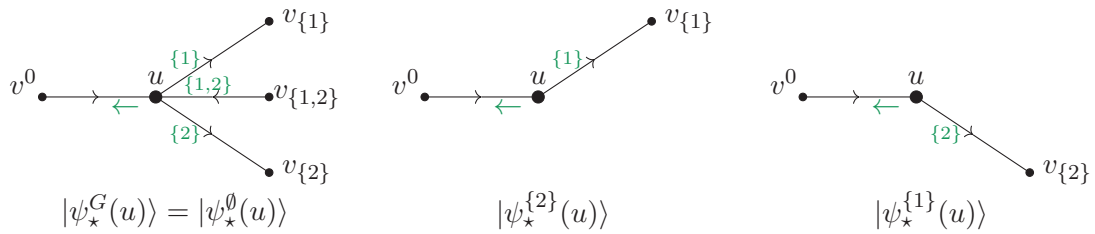


Figure 5.2: The three star states in $\Psi_\star(u)$, for $u \in V_0^+$. The edge labels from $L(u)$ are coloured. Arrows in edges indicate the direction of flow. We have chosen the flow so that flow is conserved at u in G, which can be seen by the fact that flow comes in on two edges, and leaves by two edges in the figure for $|\psi_\star^G(u)\rangle$; but flow is still conserved if we restrict to either of the other two neighbourhoods, which is necessary to satisfy Item 2 of Lemma 5.4.12.

twice as much outgoing flow as incoming flow, so the only way for flow to be conserved at u is to also have $\frac{1}{|M_0|}$ flow coming into u along edge $(v_{\{1,2\}}, u)$, so we set:

$$\theta_{u,v_{\{1,2\}}} = -\frac{1}{|M_0|}.$$

This is shown visually in Figure 5.2. Using $\mathsf{w}_1 = 1$ and $\mathsf{T}_1 = \widetilde{O}(1)$, we can compute the contribution of edges in E_1 to the energy of the flow as

$$\mathcal{R}_1 = \sum_{u \in M_0^+} \mathsf{T}_1 \frac{3(1/|M_0|)^2}{\mathsf{w}_1} = \widetilde{O}\left(\frac{|M_0^+|}{|M_0|^2}\right) = \widetilde{O}\left(\frac{|M_0|}{|M_0|^2}\right) = \widetilde{O}\left(\frac{1}{|M_0|}\right). \tag{5.28}$$

$\mathcal{R}_2$ and Item 1: Let $M_1(S)$ be the set of $v_R^1 \in V_1(S)$ such that $a_1 \in R_1(S)$ and $j^* \notin R_2$, and let $M_1 = M_1(\{1\}) \cup M_1(\{2\}) \cup M_1(\{1,2\})$, so $|M_1| = 3|M_0|$. These are exactly the vertices of V_1 that have non-zero flow coming in from V_0^+, and in particular, for $v_R^1 \in M_1(S)$, the amount of flow coming in from V_0^+ is $(-1)^{|S|}\frac{3}{|M_1|}$, and we will send it along the edge $(v_R^1, v_{R'}^2) \in E_2$ that adds j^* to the set $R_2(\mu(S))$ to obtain R':

$$\theta_{v_R^1, v_{R'}^2} = (-1)^{|S|+1}\frac{3}{|M_1|} = (-1)^{|S|+1}\frac{1}{|M_1(S)|}.$$

All other edges of E_2 will have $\theta_e = 0$. Using $\mathsf{w}_2 = \sqrt{n/m_2}$ and $\mathsf{T}_2 = \widetilde{O}(\sqrt{n/m_2})$, we can compute the contribution of edges in E_2 to the energy $\mathcal{E}^{\mathsf{T}}$ of the flow:

$$\mathcal{R}_2 = \frac{\mathsf{T}_2}{\mathsf{w}_2}|M_1|\frac{9}{|M_1|^2} = \widetilde{O}\left(\frac{1}{|M_0|}\right). \tag{5.29}$$

We also note that by ensuring that there is only flow on $v_R^1 \in V_1$ when $\mathcal{K}(R_1, A_2^{(j^*)})$ is not too big, we have ensured that the flow on the edges in $\tilde{E}$ is 0, satisfying **Item 1**.

$\mathcal{R}_3$: Finally, let $M_2(S)$ be the set of $v_R^2 \in V_2(S)$ such that $a_1 \in R_1(S)$ and $j^* \in R_2(\mu(S))$, and let $M_2 = M_2(\{1\}) \cup M_2(\{2\}) \cup M_2(\{1,2\})$. These are exactly the vertices of V_2 that have non-zero flow coming in from V_1, in the amount of $(-1)^{|S|+1}|M_2(S)|^{-1}$. We send this flow along the unique edge from v_R^2 into V_3 that adds $i_3 = a_3$:

$$\theta_{v_R^2, v_{R,a_3}^3} = (-1)^{|S|+1}\frac{1}{|M_2(S)|} = (-1)^{|S|+1}O\left(\frac{1}{|M_0|}\right).$$

Using $\mathsf{w}_3 = 1$ and $\mathsf{T}_3 = \widetilde{O}(1)$, the total contribution of edges in E_3 to the energy of the flow is:

$$\mathcal{R}_3 = \frac{\mathsf{T}_3}{\mathsf{w}_3}|M_2|O\left(\frac{1}{|M_2|^2}\right) = \widetilde{O}\left(\frac{1}{|M_0|}\right). \tag{5.30}$$

Item 5: It remains only to upper bound the energy of the flow by adding up the 4 contributions in (5.27) to (5.30), and applying $|V_0| = (1+o(1))|M_0|$ from (5.26):

$$\mathcal{E}^{\mathsf{T}}(\theta) \leq \mathcal{R}_0^+ + \mathcal{R}_1 + \mathcal{R}_2 + \mathcal{R}_3 = \widetilde{O}\left(\frac{1}{|M_0|}\right) = \widetilde{O}\left(\frac{1}{|V_0|}\right). \qquad \square$$

5.4.6 Negative analysis

For the negative analysis, we need to upper bound the total weight of the graph, taking into account the subroutine complexities, $\mathcal{W}^{\mathsf{T}}(G)$ (see Definition 3.2.7).

Lemma 5.4.13. *There exists $\mathcal{W}^{\mathsf{T}}$ such that:*

$$\mathcal{W}^{\mathsf{T}}(G) \leq \mathcal{W}^{\mathsf{T}} \leq \widetilde{O}\left(\left(n + \frac{n^2}{t_1} + \frac{n^2}{t_2}\right)|V_1|\right).$$

Proof. Recall that $\mathcal{W}^{\mathsf{T}}(G) = \mathcal{W}(G^{\mathsf{T}})$ is the total weight of the graph G^{T}, where we replace each edge e of G, with weight w_e, by a path of T_e edges of weight w_e, where T_e is the complexity of the edge transition e. Thus, $\mathcal{W}^{\mathsf{T}}(G) = \sum_{e\in E(G)} \mathsf{T}_e\mathsf{w}_e$. By Corollary 5.4.10, for all $e \in \overrightarrow{E}(G) \setminus E_2$, $\mathsf{T}_e = \widetilde{O}(1)$, and $\mathsf{w}_e = 1$ (see Table 5.2). Thus, using (5.8), the total contribution to the weight from the edges in E_0^+ is:

$$\mathcal{W}_0^+ := \mathsf{w}_0^+|E_0^+|\mathsf{T}_0^+ = \widetilde{O}\left(n|V_0|\right). \tag{5.31}$$

Using (5.13), the total contribution from the edges in E_1 is:

$$\mathcal{W}_1 := \mathsf{w}_1|E_1|\mathsf{T}_1 = \widetilde{O}\left(n|V_0|\right). \tag{5.32}$$

The edges $e \in E_2$ have $\mathsf{T}_e = \mathsf{T}_2 = \widetilde{O}(\sqrt{n/m_2})$, by Corollary 5.4.10, so using $\mathsf{w}_2 = \sqrt{n/m_2}$ and (5.15), the total contribution from the edges in E_2 is:

$$\mathcal{W}_2 := \mathsf{w}_2|E_2|\mathsf{T}_2 = \widetilde{O}\left(\sqrt{\frac{n}{m_2}}\frac{2(m_2-t_2)(n-9t_1)}{t_1+1}|V_0|\sqrt{\frac{n}{m_2}}\right) = \widetilde{O}\left(\frac{n^2}{t_1}|V_0|\right). \tag{5.33}$$

Finally, using (5.16) and the fact that $m_2 = \Theta(t_1)$, the total contribution from the edges in E_3 is:

$$\mathcal{W}_3 := \mathsf{w}_3|E_3|\mathsf{T}_3 = \widetilde{O}\left(\frac{n^2}{t_2}|V_0|\right). \tag{5.34}$$

Combining (5.31) to (5.34), we get total weight:

$$\mathcal{W}^{\mathsf{T}}(G) = \mathcal{W}_0^+ + \mathcal{W}_1 + \mathcal{W}_2 + \mathcal{W}_3 = \widetilde{O}\left(\left(n + \frac{n^2}{t_1} + \frac{n^2}{t_2}\right)|V_0|\right). \qquad \square$$

5.4.7 Conclusion of proof of Theorem 5.4.1

We can now conclude with the proof of Theorem 5.4.1, showing an upper bound of $\widetilde{O}(n^{5/7})$ on the bounded error quantum time complexity of 3-distinctness.

Proof of Theorem 5.4.1. We apply Theorem 4.3.1 to G (Section 5.4.1), M ((5.17)), σ the uniform distribution on V_0 ((5.6)), and $\Psi_\star$ (Section 5.4.2), with

$$\mathcal{W}^{\mathsf{T}} = \widetilde{O}\left(\left(n + \frac{n^2}{t_1} + \frac{n^2}{t_2}\right)|V_0|\right) \text{ and } \mathcal{R}^{\mathsf{T}} = \widetilde{O}\left(|V_0|^{-1}\right).$$

Then we have

$$\mathcal{W}^{\mathsf{T}}\mathcal{R}^{\mathsf{T}} = \widetilde{O}\left(n + \frac{n^2}{t_1} + \frac{n^2}{t_2}\right) = o(n^2).$$

We have shown the following:

Setup Subroutine: By Lemma 5.4.11, the state $|\sigma\rangle$ can be generated in cost

$$\mathsf{S} = \widetilde{O}\left(t_1 + t_2\sqrt{\frac{n}{m_2}}\right).$$

Star State Generation Subroutine: By Lemma 5.4.4, the star states $\Psi_\star$ can be generated in $\widetilde{O}(1)$ complexity.

Transition Subroutine: By Corollary 5.4.10, there is a quantum subroutine that implements the transition map with errors $\epsilon_{u,v}$ and costs $\mathsf{T}_{u,v}$ such that

TS1 For all $(u,v) \in \overrightarrow{E}(G) \setminus E_2$, $\epsilon_{u,v} = 0$. For all $(u,v) \in E_2 \setminus \tilde{E}$ (see (5.25)), taking $\kappa > 2$ in Lemma 5.4.7, we have $\epsilon_{u,v} = O(n^{-\kappa}) = o(1/(\mathcal{R}^{\mathsf{T}}\mathcal{W}^{\mathsf{T}}))$.

TS2 By Lemma 5.4.8, using $\mathsf{w}_2 = \sqrt{n/m_2}$ and $\kappa > 2$:

$$\sum_{e\in\tilde{E}} \mathsf{w}_e = \mathsf{w}_2|\tilde{E}| \le \sqrt{\frac{n}{m_2}}n^{-\kappa}|E_2| = \sqrt{\frac{n}{m_2}}n^{-\kappa}\frac{2(m_2-t_2)(n-9t_1)}{t_1+1}|V_0| \quad \text{by (5.15)}$$
$$= O\left(\sqrt{n}n^{-\kappa}n\frac{1}{\mathcal{R}^{\mathsf{T}}}\right) = o(1/\mathcal{R}^{\mathsf{T}}).$$

since $m_2 = \Theta(t_1)$.

Checking Subroutine: By (5.18), for any $u \in V_{\mathsf{M}} = V_3$, we can check if $u \in M$ in cost $\widetilde{O}(1)$.

Positive Condition: By Lemma 5.4.12, there exists a flow θ satisfying conditions **P1**-**P5** of Theorem 4.3.1, with $\mathcal{E}^{\mathsf{T}}(\theta) \le \mathcal{R}^{\mathsf{T}} = \widetilde{O}\left(|V_0|^{-1}\right)$.

Negative Condition: By Lemma 5.4.13, $\mathcal{W}^{\mathsf{T}}(G) \le \mathcal{W}^{\mathsf{T}} = \widetilde{O}\left(\left(n + \frac{n^2}{t_1} + \frac{n^2}{t_2}\right)|V_0|\right)$.

Thus, by Theorem 4.3.1, there is a quantum algorithm that decides if $M = \emptyset$ in bounded error in complexity:

$$\widetilde{O}\left(\mathsf{S} + \sqrt{\mathcal{R}^{\mathsf{T}}\mathcal{W}^{\mathsf{T}}}\right) = \widetilde{O}\left(t_1 + t_2\sqrt{\frac{n}{m_2}} + \sqrt{n + \frac{n^2}{t_1} + \frac{n^2}{t_2}}\right)$$
$$= \widetilde{O}\left(t_1 + t_2\sqrt{\frac{n}{t_1}} + \sqrt{n} + \frac{n}{\sqrt{t_1}} + \frac{n}{\sqrt{t_2}}\right).$$

Choosing the optimal values of $t_1 = n^{5/7}$ and $t_2 = n^{4/7}$, we get an upper bound of $\widetilde{O}(n^{5/7})$. Since $M \neq \emptyset$ if x has a unique 3-collision, and $M = \emptyset$ if x has no 3-collision, the algorithm distinguishes these two cases. By Lemma 5.3.1, this is enough to solve 3-distinctness in general. □

5.5 k-Distinctness algorithm

In this section, we generalise the 3-distinctness algorithm from Section 5.4 to prove the following.

Theorem 5.5.1. *Let k be any constant. There is a quantum algorithm that decides k-distinctness with bounded error in $\widetilde{O}\left(n^{\frac{3}{4}-\frac{1}{4}\frac{1}{2^k-1}}\right)$ complexity.*

We use the assumptions on the input defined in Section 5.3, including partitioning $[n]$ into $A_1 \cup \cdots \cup A_k$, and each A_ℓ, for $\ell \in \{2,\dots,k-1\}$ into blocks $A_\ell^{(1)} \cup \cdots \cup A_\ell^{(m_\ell)}$ of size $\frac{n}{km_\ell}$. A summary of the parameters of the algorithm appears in Table 5.3.

Tuples of sets: Fix constants $c_1,\dots,c_{k-1}$ and parameters $t_1,\dots,t_{k-1}$ as in Table 5.3. The vertices of our graph are labelled by sets $R = (R_1,\dots,R_{k-1})$, where (see (5.4))

$$R_1 = (R_1(S_1))_{S_1 \in 2^{[c_1]}\setminus\{\emptyset\}} \in \binom{A_1}{t_1^{(2^{c_1}-1)}}$$

is a tuple of $2^{c_1}-1$ disjoint subsets of A_1, each of size t_1, and for $\ell \in \{2,\dots,k-1\}$, R_ℓ is a tuple of $c_1 \dots c_{\ell-1}(2^{c_\ell}-1)$ disjoint subsets of $[m_\ell]$ of size t_ℓ:

$$R_\ell = (R_\ell(s_1,\dots,s_{\ell-1},S_\ell))_{s_1\in[c_1],\dots,s_{\ell-1}\in[c_{\ell-1}],S_\ell\in 2^{[c_\ell]}\setminus\{\emptyset\}}.$$

We define:

$$\overline{R}_\ell(s_1,\dots,s_{\ell-1},S_\ell) := \begin{cases} R_1(S_1) & \text{if } \ell = 1 \\ \bigcup_{j_\ell \in R_\ell(s_1,\dots,s_{\ell-1},S_\ell)} A_\ell^{(j_\ell)} & \text{if } \ell \in \{2,\dots,k-1\}, \end{cases}$$

and

$$\overline{R}_\ell := \begin{cases} R_1 & \text{if } \ell = 1 \\ (\overline{R}_\ell(s_1,\dots,s_{\ell-1},S_\ell))_{s_1\in[c_1],\dots,s_{\ell-1}\in[c_{\ell-1}],S_\ell\in 2^{[c_\ell]}\setminus\{\emptyset\}} & \text{if } \ell \in \{2,\dots,k-1\}. \end{cases}$$

If we let $r_1 = |R_1| = t_1$, and for $\ell \in \{2,\dots,k-1\}$, $r_\ell = |\overline{R}_\ell| \approx t_\ell \frac{n}{m_\ell}$, we get the set sizes r_ℓ from [Bel12a]. We will not use these variables, but we note that the values we get for $\{r_\ell\}_{\ell=1}^{k-1}$ (from the values of $\{t_\ell\}_{\ell=1}^{k-1}$) are the same as those obtained in [Bel12a], as our algorithm can be seen as an algorithmic version of the combinatorial construction used in [Bel12a].

Finally, we choose the number of blocks in each A_ℓ, m_ℓ, so that $m_\ell = \Theta(t_{\ell-1})$ for each $\ell \in \{2,\dots,k-1\}$. This ensures that the expected size of $\mathcal{K}(\overline{R}_1,\dots,\overline{R}_{\ell-1},A_\ell^{(j_\ell)})$ is constant. These values are summarised in Table 5.3.

Data: With any R defined as above, we keep track of some input-dependent data as follows. First, we query everything in R_1, so we define:

$$\begin{aligned} \forall S_1 \in 2^{[c_1]}\setminus\{\emptyset\}, D_1(R_1(S_1)) &:= \{(i_1,x_{i_1}) : i_1 \in R_1(S_1)\} \\ D_1(R) &:= (D_1(R_1(S_1)))_{S_1\in 2^{[c_1]}\setminus\{\emptyset\}}. \end{aligned} \tag{5.35}$$

Next, for $\ell \in \{2,\dots,k-1\}$, and $(s_1,\dots,s_{\ell-1},S_\ell) \in [c_1]\times\cdots\times[c_{\ell-1}]\times 2^{[c_\ell]}\setminus\{\emptyset\}$, we only query some of the indices in $\overline{R}_\ell$, and which ones we query depends on R, specifically on $R_1,\dots,R_{\ell-1}$:

$$\begin{aligned} D_\ell(R_\ell(s_1,\dots,s_{\ell-1},S_\ell)|R) := \bigcup_{\substack{S_{\ell-1}\subseteq[c_{\ell-1}]: \\ s_{\ell-1}\in S_{\ell-1}}} \big\{(i_1,\dots,i_\ell,x_{i_1}) : x_{i_\ell} = x_{i_1}, \\ i_\ell \in \overline{R}_\ell(s_1,\dots,s_{\ell-1},S_\ell), (i_1,\dots,i_{\ell-1},x_{i_1}) \in D_{\ell-1}(R_{\ell-1}(s_1,\dots,s_{\ell-2},S_{\ell-1})|R)\big\}. \end{aligned} \tag{5.36}$$

$\ell \in [k-1]$, t_ℓ	$= n^{\frac{3}{4}-\frac{1}{4}\frac{1}{2^{k}-1}-\sum_{\ell'=2}^{\ell}\frac{2^{k-1-\ell'}}{2^{k}-1}}$
$\ell \in \{2,\dots,k-1\}$ m_ℓ	$= \Theta(t_{\ell-1})$
c_1	$= k-1$
$\ell \in \{2,\dots,k-2\}$, c_ℓ	$= O(1)$ large enough for Corollary 5.5.12
c_{k-1}	$= 1$
$\ell \in \{2,\dots,k-1\}$, p_ℓ	$=$ polylog(n) large enough for Corollary 5.5.12.

Table 5.3: A summary of the (asymptotic) values of variables used in this section.

We will sometimes omit "$|R$" when the context is clear. We can group these together to get

$$D_\ell(R) := (D_\ell(R_\ell(s_1,\dots,s_{\ell-1},S_\ell)))_{(s_1,\dots,s_{\ell-1},S_\ell)\in[c_1]\times\dots[c_{\ell-1}]\times 2^{[c_\ell]}\setminus\{\emptyset\}}. \tag{5.37}$$

In addition to this data, we want to keep track of a number for each $j_\ell \in R_\ell$ that we call the *forward collision degree*. Loosely speaking, for some $i_\ell \in \overline{R}_\ell$, a forward collision is an element $(i_1,\dots,i_\ell,\dots,i_{\ell'},x_{i_1}) \in D_{\ell'}(R)$, for some $\ell' > \ell$, and some $i_1,\dots,i_{\ell-1},i_{\ell+1},\dots,i_{\ell'}$. This can only exist if $(i_1,\dots,i_\ell,i_{\ell+1},x_{i_1}) \in D_{\ell+1}(R)$, so the *forward collision degree of* i_ℓ, $\bar{d}_\ell^{\rightarrow}(i_\ell)$, counts these:

$$\bar{d}_R^{\rightarrow}(i_\ell) := \big|\big\{(i_1,\dots,i_{\ell-1},i_{\ell+1}) \in \overline{R}_1\times\dots\times\overline{R}_{\ell-1}\times\overline{R}_{\ell+1} : (i_1,\dots,i_\ell,i_{\ell+1},x_{i_1}) \in D_{\ell+1}(R)\big\}\big|. \tag{5.38}$$

Then for $\ell \in \{2,\dots,k-2\}$, we can define the forward collision degree of $j_\ell \in R_\ell$ as

$$d_R^{\rightarrow}(j_\ell) := \sum_{i_\ell \in A_\ell^{(j_\ell)}} \bar{d}_\ell^{\rightarrow}(i_\ell). \tag{5.39}$$

For consistency, we also define $d_R^{\rightarrow}(i_1) := \bar{d}_R^{\rightarrow}(i_1)$ for $i_1 \in R_1$, and $d_R^{\rightarrow}(j_{k-1}) := 0$ for $j_{k-1} \in R_{k-1}$. When our quantum walk removes some j_ℓ from R_ℓ, we will want to make sure that $d_R^{\rightarrow}(j_\ell) = 0$, because otherwise we will have to uncompute all forward collisions from the data, which could be expensive. Thus, we also keep a database of forward collision degrees:

$$\forall \ell \in [k-2],\ C_\ell^{\rightarrow}(R) := \{(j_\ell, d_R^{\rightarrow}(j_\ell)) : j_\ell \in R_\ell, d_R^{\rightarrow}(j_\ell) > 0\}. \tag{5.40}$$

To summarise, the data we keep track of at a vertex v_R includes:

$$D(R) := (D_1(R),\dots,D_{k-1}(R),C_1^{\rightarrow}(R),\dots,C_{k-2}^{\rightarrow}(R)). \tag{5.41}$$

5.5.1 The graph: vertex sets

To define G, we begin by defining disjoint vertex sets V_0, V_0^+, $(V_\ell)_{\ell=1}^{k-1}$, $(V_\ell^+)_{\ell=1}^{k-2}$, and V_k, whose union makes up $V(G)$. We will use the notation in (5.4) and (5.5) for tuples of disjoint sets throughout this section. Figure 5.3 summarises G.

V_0: We define

$$V_0 = \Bigg\{ v^0_{R_1,\dots,R_{k-1}} := (0, R_1, \dots, R_{k-1}, D(R_1,\dots,R_{k-1})) : \\ R_1 \in \binom{A_1}{t_1^{(2^{c_1}-1)}} \text{ and } \forall \ell \in \{2,\dots,k-1\},\ R_\ell \in \binom{[m_\ell]}{t_\ell^{(c_1\cdots c_{\ell-1}(2^{c_\ell}-1))}} \Bigg\}. \quad (5.42)$$

Our initial distribution is uniform on V_0: $\sigma(u) = \frac{1}{|V_1|}$ for all $u \in V_0$. We implicitly store all sets including those making up $R_1,\dots,R_{k-1}$ and $D(R_1,\dots,R_{k-1})$ in a data structure with the properties described in Section 3.5.1. This will only be important when we analyze the time complexity of the setup and transition subroutines.

V_0^+: At a vertex in V_0^+, we suppose we have chosen a new element i_1 to add to R_1, but not yet added it. Thus, we label such a vertex by a tuple of sets R, and an index $i_1 \notin R_1$:

$$V_0^+ := \Big\{ v^0_{R_1,\dots,R_{k-1},i_1} := ((0,+), R_1, \dots, R_{k-1}, D(R_1,\dots,R_{k-1}), i_1) : \\ v^0_{R_1,\dots,R_{k-1}} \in V_0, i_1 \in A_1 \setminus R_1 \Big\},$$

$$\text{so } \left|V_0^+\right| = |V_0|\,|A_1 \setminus R_1| = O\left(n\,|V_0|\right). \quad (5.43)$$

V_ℓ for $\ell \in [k-1]$: At a vertex in V_ℓ, we suppose we have added a new element to each of $R_1,\dots,R_\ell$, meaning that for each $\ell' \in [\ell]$, there is some $(s_1,\dots,s_{\ell'-1},S_{\ell'}) \in [c_1] \times \cdots \times [c_{\ell'-1}] \times (2^{[c_{\ell'}]} \setminus \{\emptyset\})$ such that $|R_{\ell'}(s_1,\dots,s_{\ell'-1},S_{\ell'})| = t_{\ell'}+1$. However, we will not let the choices of $s_1,\dots,s_{\ell'-1}$ for different ℓ' be arbitrary. Instead, we define the following sets of vertices, for $(S_1,\dots,S_\ell) \in (2^{[c_1]} \setminus \{\emptyset\}) \times \cdots \times (2^{[c_\ell]} \setminus \{\emptyset\})$, where $\mu(S)$ denotes the minimum element of a set S:

$$V_\ell(S_1,\dots,S_\ell) := \Big\{ v^\ell_R = (\ell, R, D(R)) : R_1 \in \binom{A_1}{t_1^{(2^{c_1}-1)}}^+; \\ \forall \ell' \in \{2,\dots,\ell\}, R_{\ell'} \in \binom{[m_\ell]}{t_\ell^{(c_1\dots c_{\ell-1}(2^{c_\ell}-1))}}^+; \\ \forall \ell' \in \{\ell+1,\dots,k-1\}, R_{\ell'} \in \binom{[m_\ell]}{t_\ell^{(c_1\dots c_{\ell-1}(2^{c_\ell}-1))}}; \\ \forall \ell' \in [\ell],\ |R_{\ell'}(\mu(S_1),\dots,\mu(S_{\ell'-1}),S_{\ell'})| = t_{\ell'}+1 \Big\}. \quad (5.44)$$

This is the set of vertices labelled by sets R where we have added elements to each of $R_1,\dots,R_\ell$, not yet added elements to $R_{\ell+1},\dots,R_{k-1}$, and for $\ell' \in [\ell]$, the choice of *where* the new element was added to $R_{\ell'}$ is determined by $S_1,\dots,S_\ell$. Then we can define:

$$V_\ell := \bigcup_{(S_1,\dots,S_\ell) \in (2^{[c_1]}\setminus\{\emptyset\}) \times \cdots \times (2^{[c_\ell]}\setminus\{\emptyset\})} V_\ell(S_1,\dots,S_\ell). \quad (5.45)$$

Using the fact that for all $\ell' \in \{2,\dots,k-2\}$, $m_{\ell'} = \Theta(t_{\ell'-1})$, we have

$$|V_\ell| = O\left(|V_0|\,\frac{n}{t_1}\prod_{\ell'=2}^{\ell}\frac{m_{\ell'}}{t_{\ell'}}\right) = O\left(|V_0|\,\frac{n}{t_1}\prod_{\ell'=2}^{\ell}\frac{t_{\ell'-1}}{t_{\ell'}}\right) = O\left(\frac{n}{t_\ell}\,|V_0|\right). \quad (5.46)$$

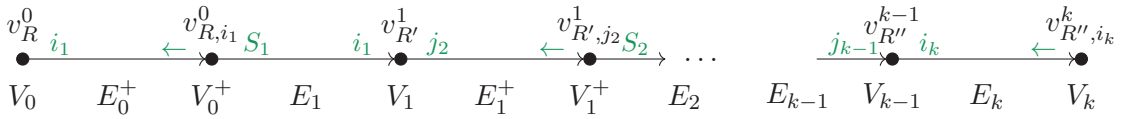


Figure 5.3: A path from V_0 to V_k, with edge labels shown in coloured. R' is obtained from R by inserting i_1 into $R_1(S_1)$. R'' is obtained from R' by inserting j_2 into $R_2(\mu(S_1), S_2)$, and for some choice of $S_3, \dots, S_{k-1}$, inserting, for each $\ell \in \{3, \dots, k-1\}$, some j_ℓ into $R_\ell(\mu(S_1), \dots, \mu(S_{\ell-1}), S_\ell)$.

V_ℓ^+ : **for** $\ell \in [k-2]$**:** At a vertex in V_ℓ^+, we suppose, as in V_ℓ, that we have already added an element to each of the sets $R_1, \dots, R_\ell$, but now have also selected an element $j_{\ell+1} \in [m_{\ell+1}]$ to add to $R_{\ell+1}$:

$$V_\ell^+(S_1, \dots, S_\ell) := \Big\{ v_{R,j_{\ell+1}}^\ell := ((\ell,+), R, D(R), j_{\ell+1}) : v_R^\ell \in V_\ell(S_1, \dots, S_\ell), j_{\ell+1} \in [m_{\ell+1}] \setminus R_{\ell+1} \Big\}$$

$$V_\ell^+ := \bigcup_{(S_1,\dots,S_\ell) \in (2^{[c_1]} \setminus \{\emptyset\}) \times \dots \times (2^{[c_\ell]} \setminus \{\emptyset\})} V_\ell^+(S_1, \dots, S_\ell),$$

so together with (5.46) and $m_{\ell+1} = \Theta(t_\ell)$, this implies

$$\left|V_\ell^+\right| = |V_\ell| \, |[m_{\ell+1}] \setminus T_{\ell+1}| = O\left(n \, |V_0|\right). \tag{5.47}$$

The final stage, V_k**:** At a vertex in V_k, we have added a new element to each of the sets $R_1, \dots, R_{k-1}$, as in V_{k-1}, and also selected some $i_k \in A_k$, which we can view as a candidate for completing one of the $(k-1)$-collisions in $D_{k-1}(R)$ to a k-collision:

$$\begin{aligned} V_k &:= \left\{ v_{R,i_k}^k := (k, R, D(R), i_k) : v_R^{k-1} \in V_{k-1}, i_k \in A_k. \right\}, \\ \text{so } |V_k| &= |V_{k-1}| \, |A_k| = O\left(\frac{n^2}{t_{k-1}} \, |V_0| \right). \end{aligned} \tag{5.48}$$

5.5.2 The graph: edge sets

We now define the sets of edges that make up $\overrightarrow{E}(G)$, as well as the edge label sets $L(u)$ (see Definition 3.2.13) for each $u \in V(G)$. These are also summarised in Table 5.4.

$E_0^+ \subset V_0 \times V_0^+$**:** There is an edge between $v_R^0 \in V_0$ and $v_{R,i_1}^0 \in V_0^+$ for any $i_1 \in A_1 \setminus R_1$, and for any $v_{R,i_1}^0 \in V_0^+$, v_R^0 is its unique in-neighbour (in V_0), so we define

$$L^+(v_R^0) := A_1 \setminus R_1 \text{ and } L^-(v_{R,i_1}^0) := \{\leftarrow\},$$

and let $f_{v_R^0}(i_1) = v_{R,i_1}^0$, and $f_{v_{R,i_1}^0}(\leftarrow) = v_R^0$. We let E_0^+ be the set of all such edges

$$E_0^+ := \left\{ \left(v_R^0, v_{R,i_1}^0\right) : v_R^0 \in V_0, i_1 \in A_1 \setminus R_1 \right\}$$

and set $\mathsf{w}_e = \mathsf{w}_0^+ = 1$ for all $e \in E_0^+$. This together with (5.43) implies that

$$\left|E_0^+\right| = \left|V_0^+\right| = O\left(n \, |V_0|\right). \tag{5.49}$$

u	$j \in L^-(u)$	$f_u(j)$	$i \in L^+(u)$	$f_u(i)$
$v_R^0 \in V_0$	$\emptyset$		$i_1 \in A_1 \setminus R_1$	v_{R,i_1}^0
$v_{R,i_1}^0 \in V_0^+$	$\leftarrow$	v_R^0	$S_1 \in 2^{[c_1]} \setminus \{\emptyset\}$	$v_{R^{S_1 \leftarrow i_1}}^1$
$v_R^\ell \in V_\ell(S)$	$j_\ell \in R_\ell(\hat{\mu}(S)) :$ $d_R^{\rightarrow}(j_\ell) = 0$	$v_{R\setminus\{j_\ell\},j_\ell}^{\ell-1}$	$j_{\ell+1} \in [m_{\ell+1}] \setminus R_{\ell+1}$	$v_{R,j_{\ell+1}}^\ell$
$v_{R,j_{\ell+1}}^{\ell+1} \in V_{\ell+1}^+$	$\leftarrow$	$v_R^{\ell+1}$	$S_{\ell+1} \in 2^{[c_{\ell+1}]} \setminus \{\emptyset\}$	$v_{R^{S_{\ell+1} \leftarrow j_{\ell+1}}}^{\ell+1}$
$v_R^{k-1} \in V_{k-1}(S)$	$j_{k-1} \in R_{k-1}(\hat{\mu}(S))$	$v_{R\setminus\{j_{k-1}\},j_{k-1}}^{k-2}$	$i_k \in A_k$	v_{R,i_k}^k
$v_{R,i_k}^k \in V_k$	$\leftarrow$	v_R^{k-1}	$\emptyset$	

Table 5.4: The sets labelling incoming (L^-) and outgoing (L^+) edges of each vertex $u \in V(G)$, and the neighbouring vertices at the end of every such edge. $\ell \in [k-2]$, $S = (S_1, \ldots, S_\ell)$, and for brevity we use $\hat{\mu}(S) := (\mu(S_1), \ldots, \mu(S_{\ell-1}), S_\ell)$, where μ is the minimum. $R^{S_1 \leftarrow i_1}$ is obtained from R by inserting i_1 into $R_1(S_1)$, and for $v_{R,j_{\ell+1}}^\ell \in V_{\ell+1}^+(S)$, $R^{S_{\ell+1} \leftarrow j_{\ell+1}}$ is obtained from R by inserting $j_{\ell+1}$ into $R_{\ell+1}(\hat{\mu}(S))$. To ensure that $L^-(u)$ and $L^+(u)$ are always disjoint, we implicitly append a $\leftarrow$ label to all of $L^-(u)$ and a $\rightarrow$ label to all of $L^+(u)$.

$E_\ell^+ \subset V_\ell \times V_\ell^+$ for $\ell \in [k-2]$: There is an edge between $v_R^\ell \in V_\ell$ and $v_{R,j_{\ell+1}}^\ell \in V_\ell^+$ for any $j_{\ell+1} \in [m_{\ell+1}] \setminus R_{\ell+1}$, so we define

$$L^+(v_R^\ell) := [m_{\ell+1}] \setminus R_{\ell+1} \text{ and } L^-(v_{R,j_{\ell+1}}^\ell) := \{\leftarrow\},$$

and let $f_{v_R^\ell}(j_{\ell+1}) = v_{R,j_{\ell+1}}^\ell$ and $f_{v_{R,j_{\ell+1}}^\ell}(\leftarrow) = v_R^\ell$. We let E_ℓ^+ be the set of all such edges

$$E_\ell^+ := \left\{(v_R^\ell, v_{R,j_{\ell+1}}^\ell) : v_R^\ell \in V_\ell, j_{\ell+1} \in [m_{\ell+1}] \setminus R_{\ell+1}\right\}$$

and set $\mathsf{w}_e = \mathsf{w}_\ell^+ = 1$ for all $e \in E_\ell^+$. This together with (5.47) implies that

$$\left|E_\ell^+\right| = \left|V_\ell^+\right| = O\left(n\left|V_0\right|\right). \tag{5.50}$$

Faults: Fix $\ell \in [k-1]$. As in the case of 3-distinctness, if we add a new element j_ℓ to certain parts of R_ℓ, to get R', such that $d_{R'}^{\rightarrow}(j_\ell) > 0$, this introduces a *fault* in the data, which our quantum walk will want to avoid. The case for $k > 3$ is slightly more complicated, so we examine exactly when a fault is introduced before describing the remaining edge sets.

Suppose we are at the vertex $v_R^{\ell-1} \in V_{\ell-1}(S_1^*, \ldots, S_{\ell-1}^*)$ (see (5.44)) and we want to add to the set $\overline{R}_\ell(\mu(S_1^*), \ldots, \mu(S_{\ell-1}^*), S_\ell)$, for some $S_\ell \subseteq [c_\ell]$, an index i_ℓ (by adding j_ℓ such that $i_\ell \in A_\ell^{(j_\ell)}$ to $R_\ell(\mu(S_1^*), \ldots, \mu(S_{\ell-1}^*), S_\ell)$). For $\ell \in \{2, \ldots, k-2\}$, this introduces a fault if the following conditions are satisfied, where we use $[\mathcal{E}]$ to denote the logical value of an event $\mathcal{E}$:

$$\begin{aligned}
\mathbf{C}^{\leftarrow}(i_\ell, R, S_\ell) &:= \big[\exists (i_1, \ldots, i_{\ell-1}) \in R_1 \times \overline{R}_2 \times \cdots \times \overline{R}_{\ell-1} \text{ s.t.} \\
&\qquad\qquad (i_1, \ldots, i_{\ell-1}, i_\ell, x_{i_1}) \in D_\ell(R_\ell(\mu(S_1^*), \ldots, \mu(S_{\ell-1}^*), S_\ell))\big] \\
\mathbf{C}^{\rightarrow}(i_\ell, R, S_\ell) &:= \Bigg[\exists s_\ell \in S_\ell \text{ s.t.} \\
&\qquad \exists i_{\ell+1} \in \bigcup_{S_{\ell+1} \in 2^{[c_{\ell+1}]} \setminus \{\emptyset\}} \overline{R}_{\ell+1}(\mu(S_1^*), \ldots, \mu(S_{\ell-1}^*), s_\ell, S_{\ell+1}) \text{ s.t. } x_{i_{\ell+1}} = x_{i_\ell}\Bigg].
\end{aligned} \tag{5.51}$$

In words, $\mathbf{C}^{\leftarrow}$ is the condition that i_ℓ forms a collision $(i_1, \ldots, i_\ell, x_{i_1})$ that would be stored in $D_\ell(R)$, and $\mathbf{C}^{\rightarrow}$ is the condition that i_ℓ collides with something in $\overline{R}_{\ell+1}$ such that if $\mathbf{C}^{\rightarrow}$ holds, $(i_1, \ldots, i_{\ell+1}, x_{i_1})$ would be stored in $D_{\ell+1}(R)$. For $\ell = 1$, $\mathbf{C}^{\rightarrow}$ is also defined, and i_1 introduces a fault whenever $\mathbf{C}^{\rightarrow}$ is true. For $\ell = k-1$, $\mathbf{C}^{\rightarrow}$ can never be true, so there is never a fault. We set $c_{k-1} = 1$ (see Table 5.3).

Then for any $\ell \in [k-2]$, $v_R^{\ell-1} \in V_{\ell-1}(S_1^*, \ldots, S_{\ell-1}^*)$, $i_\ell \in A_\ell \setminus \overline{R}_\ell$, and $S_\ell \in 2^{[c_\ell]} \setminus \{\emptyset\}$, condition $\mathbf{C}^{\rightarrow}$ is false if and only if S_ℓ is disjoint from the following set:

$$\mathcal{I}(v_R^{\ell-1}, i_\ell) := \Bigg\{ s_\ell \in [c_\ell] : \exists i_{\ell+1} \in \bigcup_{S_{\ell+1} \in 2^{[c_{\ell+1}]} \setminus \{\emptyset\}} \overline{R}_{\ell+1}(\mu(S_1^*), \ldots, \mu(S_{\ell-1}^*), s_\ell, S_{\ell+1}) \text{ s.t. } x_{i_{\ell+1}} = x_{i_\ell} \Bigg\}. \tag{5.52}$$

For $\ell = k-1$, we define $\mathcal{I}(v_R^{k-2}, i_{k-1}) := \emptyset$. When $\ell = 1$ we can define, for $v_{R,i_1}^0 \in V_0^+$:

$$\mathcal{I}(v_{R,i_1}^0) := \mathcal{I}(v_R^0, i_1).$$

As long as we choose some S_1 that avoids this set, we will not introduce a fault. For $\ell > 1$, examining condition $\mathbf{C}^{\leftarrow}$ above, although it appears to depend on S_ℓ, it does not. Referring to (5.36), we can rewrite $\mathbf{C}^{\leftarrow}$ as

$$\mathbf{C}^{\leftarrow}(i_\ell, R, S_\ell) \Leftrightarrow \mathbf{C}^{\leftarrow}(i_\ell, R) := \Big[\exists S_{\ell-1} \subseteq [c_{\ell-1}] \text{ s.t. } \mu(S_{\ell-1}^*) \in S_{\ell-1}, \exists (i_1, \ldots, i_{\ell-1}, x_{i_1}) \in D_{\ell-1}(R_{\ell-1}(\mu(S_1^*), \ldots, \mu(S_{\ell-2}^*), S_{\ell-1})) \text{ s.t. } x_{i_\ell} = x_{i_1} \Big].$$

Thus, for $\ell \in \{2, \ldots, k-2\}$, for any $v_{R,j_\ell}^{\ell-1} \in V_{\ell-1}^+$, we can define:

$$\mathcal{I}(v_{R,j_\ell}^{\ell-1}) := \bigcup_{i_\ell \in A_\ell^{(j_\ell)} : \mathbf{C}^{\leftarrow}(i_\ell, R)} \mathcal{I}(v_R^{\ell-1}, i_\ell). \tag{5.53}$$

Lemma 5.5.2. *Fix any $v_{R,i_1}^0 \in V_0^+$ and non-empty $S_1 \subseteq [c_1]$. Insert i_1 into $R_1(S_1)$ to obtain $R' = (R_1^{S_1 \leftarrow i_1}, R_2, \ldots, R_{k-1})$. Then $d_{R'}^{\rightarrow}(i_1) = 0$ if and only if $S_1 \cap \mathcal{I}(v_{R,i_1}^0) = \emptyset$.*

Similarly, for any $\ell \in \{2, \ldots, k-1\}$, fix $v_{R,j_\ell}^{\ell-1} \in V_{\ell-1}^+(S_1^*, \ldots, S_{\ell-1}^*)$, and non-empty $S_\ell \subseteq [c_\ell]$, and let R' be obtained from R by inserting j_ℓ into $R_\ell(\mu(S_1^*), \ldots, \mu(S_{\ell-1}^*), S_\ell)$. Then $d_{R'}^{\rightarrow}(j_\ell) = 0$ if and only if $S_\ell \cap \mathcal{I}(v_{R,j_\ell}^{\ell-1}) = \emptyset$.

Proof. For $\ell = k-1$, $d_{R'}^{\rightarrow}(j_{k-1}) = 0$ and $\mathcal{I}(v_{R,j_{k-1}}^{k-2}) = \emptyset$ always hold, by definition. By the definition of $\bar{d}_{R'}^{\rightarrow}(i_\ell)$ from (5.38) we have for $\ell \in [k-2]$,

$$\begin{aligned}
\bar{d}_{R'}^{\rightarrow}(i_\ell) &= |\{(i_1, \ldots, i_\ell, i_{\ell+1}, x_{i_1}) \in D_{\ell+1}(R')\}| \\
&= \sum_{\substack{(s_1, \ldots, s_\ell, S_{\ell+1}) \in \\ [c_1] \times \cdots \times [c_\ell] \times (2^{[c_{\ell+1}]} \setminus \{\emptyset\})}} |\{(i_1, \ldots, i_{\ell+1}, x_{i_1}) \in D_{\ell+1}(R(s_1, \ldots, s_\ell, S_{\ell+1}))\}| \\
&= \sum_{\substack{(s_1, \ldots, s_\ell, S_{\ell+1}) \in \\ [c_1] \times \cdots \times [c_\ell] \times (2^{[c_{\ell+1}]} \setminus \{\emptyset\})}} \sum_{\substack{S'_\ell \subseteq [c_\ell]: \\ s_\ell \in S_\ell}} |\{(i_1, \ldots, i_{\ell+1}, x_{i_1}) : i_{\ell+1} \in \overline{R}_{\ell+1}(s_1, \ldots, s_\ell, S_{\ell+1}), \\
&\qquad x_{i_{\ell+1}} = x_{i_\ell}, (i_1, \ldots, i_\ell, x_{i_1}) \in D_\ell(R'_\ell(s_1, \ldots, s_{\ell-1}, S'_\ell))\}|,
\end{aligned}$$

where we have used (5.37) and (5.36). If $\ell = 1$, let i_ℓ be the i_1 from the lemma statement. Otherwise, suppose $i_\ell \in A_\ell^{(j_\ell)}$. In either case, we have $i_\ell \in \overline{R}_\ell(\mu(S_1^*), \ldots, \mu(S_{\ell-1}^*), S_\ell)$, so

by (5.36), ℓ-collisions of the form $(i_1, \dots, i_\ell, x_{i_1})$ can only occur in the data $D_\ell(R_\ell(\mu(S_1^*), \dots, \mu(S_{\ell-1}^*), S_\ell))$, so we continue:

$$\bar{d}^{\rightarrow}_{R'}(i_\ell) = \sum_{\substack{s_\ell \in S_\ell, \\ S_{\ell+1} \in 2^{[c_{\ell+1}]} \setminus \{\emptyset\}}} |\{(i_1, \dots, i_{\ell+1}, x_{i_1}) : i_{\ell+1} \in \overline{R}_{\ell+1}(\mu(S_1^*), \dots, \mu(S_{\ell-1}^*), s_\ell, S_{\ell+1}), \; x_{i_{\ell+1}} = x_{i_\ell}, (i_1, \dots, i_\ell, x_{i_1}) \in D_\ell(R'_\ell(\mu(S_1^*), \dots, \mu(S_{\ell-1}^*), S_\ell))\}|,$$

and thus, $\bar{d}^{\rightarrow}_{R'}(i_\ell) > 0$ if and only if:

$$\exists s_\ell \in S_\ell, S_{\ell+1} \in 2^{[c_{\ell+1}]} \setminus \{\emptyset\} \text{ s.t. } \exists i_{\ell+1} \in \overline{R}_{\ell+1}(\mu(S_1^*), \dots, \mu(S_{\ell-1}^*), s_\ell, S_{\ell+1}) \text{ s.t. } x_{i_{\ell+1}} = x_{i_\ell} \tag{5.54}$$

$$\text{and } \exists (i_1, \dots, i_{\ell-1}, i_\ell, x_{i_1}) \in D_\ell(R'_\ell(\mu(S_1^*), \dots, \mu(S_{\ell-1}^*), S_\ell)). \tag{5.55}$$

The first condition is exactly $\mathbf{C}^{\rightarrow}(i_\ell, R, S_\ell)$ (see (5.51)), which is satisfied if and only if $S_\ell \cap \mathcal{I}(v_R^0, i_1) = \emptyset$ (see (5.52)). In the case $\ell = 1$, the second condition is just $(i_1, x_{i_1}) \in D_1(R'_1(S_1))$, which is true by (5.35), since we just added i_1 to $R_1(S_1)$ to get $R'(S_1)$. This completes the $\ell = 1$ case, since $d^{\rightarrow}_{R'}(i_1) = \bar{d}^{\rightarrow}_{R'}(i_1)$, and $\mathcal{I}(v^0_{R,i_1}) = \mathcal{I}(v^0_R, i_1)$.

Continuing with the case $\ell \in \{2, \dots, k-2\}$, suppose $d^{\rightarrow}_{R'}(j_\ell) > 0$. By (5.39), this happens if and only if there exists $i_\ell \in A_\ell^{(j_\ell)}$ such that $\bar{d}^{\rightarrow}_{R'}(i_\ell) > 0$, which holds if and only if (5.54) and (5.55) are true. We know (5.54) if and only if $\mathbf{C}^{\rightarrow}(i_\ell, R, S_\ell)$ holds, if and only if $S_\ell \cap \mathcal{I}(v_R^{\ell-1}, i_\ell) \neq \emptyset$. By (5.36), using the fact that $R'_{\ell-1} = R_{\ell-1}$, we have (5.55) if and only if $\mathbf{C}^{\leftarrow}(i_\ell, R)$. Thus, we have

$$[d^{\rightarrow}_{R'}(j_\ell) > 0] \Leftrightarrow \underbrace{\exists i_\ell \in A_\ell^{(j_\ell)} \text{ s.t } \left[\left[S_\ell \cap \mathcal{I}(v_R^{\ell-1}, i_\ell) \neq \emptyset\right] \wedge \mathbf{C}^{\leftarrow}(i_\ell, R)\right]}_{=:\mathbf{C}}.$$

If $\mathbf{C}$ holds, then then by (5.53), $\mathcal{I}(v_R^{\ell-1}, i_\ell) \subseteq \mathcal{I}(v_{R,j_\ell}^{\ell-1})$, and so, also by $\mathbf{C}$, $S_\ell \cap \mathcal{I}(v_{R,j_\ell}^{\ell-1}) \neq \emptyset$. For the other direction, if $S_\ell \cap \mathcal{I}(v_{R,j_\ell}^{\ell-1}) \neq \emptyset$, then by (5.53), $\exists i_\ell \in A_\ell^{(j_\ell)}$ satisfying both conditions of $\mathbf{C}$. □

$E_1 \subset V_0^+ \times V_1$: Recall that V_0^+ is the set of vertices v^0_{R,i_1} in which we have chosen an index i_1 to add to R_1, but not yet decided to which part of R_1 it should be added. A transition in E_1 represents selecting some $S_1 \in 2^{[c_1]} \setminus \{\emptyset\}$ and then adding i_1 to $R_1(S_1)$, so we have

$$L^+(v^0_{R,i_1}) := 2^{[c_1]} \setminus \{\emptyset\},$$

and $f_{v^0_{R,i_1}}(S_1) = v^1_{R'}$, where R' is obtained from R by inserting i_1 into $R_1(S_1)$. As in the case of 3-distinctness, not all of these labels represent edges with non-zero weight. To go from a vertex $v^1_{R'} \in V_1(S_1^*)$, we choose some i_1 to remove from $R'_1(S_1^*)$, the part of R_1 that has had an index added, to get some R such that $v^0_R \in V_0$. However, we make sure to choose an i_1 with no forward collisions – i.e. $d^{\rightarrow}_{R'}(i_1) = 0$ – so we let

$$L^-(v^1_{R'}) := \{i_1 \in R'_1(S_1^*) : d^{\rightarrow}_{R'}(i_1) = 0\},$$

and then set $f_{v^1_{R'}}(i_1) = v^0_{R,i_1}$ where $R = R' \setminus i_1$ is obtained from R' by removing i_1. Importantly, given $v^1_{R'}$, we can take a superposition over this set, because we store the set $C_1^{\rightarrow}(R)$ defined in (5.40) (this is necessary in Section 5.5.3).

As in the case of 3-distinctness, it is not yet clear how to define E_1, the set of (non-zero weight) edges between V_0^+ and V_1, because $|V_0^+| \cdot |L^+(v^0_{r,i_1})| > |V_1| \cdot |L^-(v^1_{R'})|$. We define it as follows.

$$E_1 := \left\{\left(f_{v^1_{R'}}(i_1), v^1_{R'}\right) = \left(v^0_{R' \setminus \{i_1\}, i_1}, v^1_{R'}\right) : v^1_{R'} \in V_1, i_1 \in L^-(v^1_{R'})\right\}$$

and give weight $\mathsf{w}_1 = 1$ to all edges in E_1. Then we have the following.

Lemma 5.5.3. *Let $R^{S_1 \leftarrow i_1}$ be obtained from R by inserting i_1 into $R_1(S_1)$. Then*

$$E_1 = \left\{ \left(v^0_{R,i_1}, v^1_{R^{S_1 \leftarrow i_1}}\right) : v^0_{R,i_1} \in V_0^+, S_1 \in 2^{[c_1] \setminus \mathcal{I}(v^0_{R,i_1})} \setminus \{\emptyset\} \right\}.$$

So for all $v^0_{R,i_1} \in V_0^+$, and $S_1 \in L^+(v^0_{R,i_1})$, $\mathsf{w}_{v^0_{R,i_1},S_1} = \begin{cases} \mathsf{w}_1 = 1 & \text{if } S_1 \cap \mathcal{I}(v^0_{R,i_1}) = \emptyset \\ 0 & \text{else.} \end{cases}$

Proof. Let E_1' be the right-hand side of the identity in the theorem statement, so we want to show $E_1 = E_1'$. Fix any $v^0_{R,i_1} \in V_0^+$ and non-empty $S_1 \subseteq [c_1] \setminus \mathcal{I}(v_{R,i_1})$, and let $R' = R^{S_1 \leftarrow i_1}$. Then since $S_1 \cap \mathcal{I}(v_{R,i_1}) = \emptyset$, by Lemma 5.5.2, $d^{\rightarrow}_{R'}(i_1) = 0$. This implies $E_1' \subseteq E_1$.

For the other direction, fix any $v^1_{R'} \in V_1(S_1^*)$ and $i_1 \in L^-(v^1_{R'})$. Since $i_1 \in R_1(S_1^*)$, we have $v^1_{R' \setminus \{i_1\}} \in V_0^+$ (that is, we have removed an index from the set that had size $t_1 + 1$) and $(R' \setminus \{i_1\})^{S_1^* \leftarrow i_1} = R'$. Since $d^{\rightarrow}_{R'}(i_1) = 0$, by Lemma 5.5.2, $S_1^* \cap \mathcal{I}(v^0_{R' \setminus \{i_1\}, i_1}) = \emptyset$. This implies $E_1 \subseteq E_1'$. □

We remark that for any $i_1 \in A_1$, $d^{\rightarrow}_R(i_1)$ is always at most $k-2$. Otherwise, there are at least $k-1$ elements $i_2 \in \overline{R}_2 \subset A_2$ such that $x_{i_1} = x_{i_2}$, and together with i_1 these form a k-collision, which contradicts our assumption that the unique k-collision is in $A_1 \times \cdots \times A_k$. Thus, if we set $c_1 = k-1$, we have for any $v^0_{R,i_1} \in V_0^+$, $\mathcal{I}(v^0_{R,i_1}) \subsetneq [c_1]$, which will be important in Section 5.5.3.

Finally, it follows from (5.43), that

$$|E_1| \leq \left|L^+(v^0_{R,i_1})\right| \left|V_0^+\right| = O\left(n\,|V_0|\right). \tag{5.56}$$

$E_\ell \subset V_{\ell-1}^+ \times V_\ell$ for $\ell \in \{2, \ldots, k-1\}$: Similar to the definition E_1, we define, for any $v^{\ell-1}_{R,j_\ell} \in V^+_{\ell-1}$, and $v^\ell_{R'} \in V_\ell(S_1^*, \ldots, S_\ell^*)$:

$$L^+(v^{\ell-1}_{R,j_\ell}) := 2^{[c_\ell]} \setminus \{\emptyset\} \text{ and } L^-(v^\ell_{R'}) := \{j_\ell \in R_\ell(\mu(S_1^*), \ldots, \mu(S_{\ell-1}^*), S_\ell) : d^{\rightarrow}_R(j_\ell) = 0\}.$$

We set $f_{v^{\ell-1}_{R,j_\ell}}(S_\ell) = v^\ell_{R'}$ where if $v^{\ell-1}_R \in V_{\ell-1}(S_1^*, \ldots, S_{\ell-1}^*)$, R' is obtained from R by inserting j_ℓ into $R_\ell(\mu(S_1^*), \ldots, \mu(S_{\ell-1}^*), S_\ell)$. We set $f_{v^\ell_{R'}}(j_\ell) = v^{\ell-1}_{R' \setminus \{j_\ell\}, j_\ell}$. Similar to E_1, we define:

$$E_\ell := \left\{ \left(f_{v^\ell_{R'}}(j_\ell), v^\ell_{R'}\right) = \left(v^{\ell-1}_{R' \setminus \{j_\ell\}, j_\ell}, v^\ell_{R'}\right) : v^\ell_{R'} \in V_\ell, j_\ell \in L^-(v^\ell_{R'}) \right\}, \tag{5.57}$$

and give weight $\mathsf{w}_\ell := \sqrt{n/m_{\ell-1}}$ to all edges in E_ℓ. Then we have the following.

Lemma 5.5.4. *For any $(S_1, \ldots, S_{\ell-1}) \in (2^{[c_1]} \setminus \{\emptyset\}) \times \cdots \times (2^{[c_{\ell-1}]} \setminus \{\emptyset\})$, define:*

$$E_\ell(S_1, \ldots, S_{\ell-1}) = \left\{ \left(v^{\ell-1}_{R,j_\ell}, v^\ell_{R'}\right) : v^{\ell-1}_{R,j_\ell} \in V^+_{\ell-1}(S_1, \ldots, S_{\ell-1}), \right.$$
$$\left. \exists S_\ell \in 2^{[c_\ell] \setminus \mathcal{I}(v^{\ell-1}_{R,j_\ell})} \setminus \{\emptyset\},\ R' = R^{(\mu(S_1), \ldots, \mu(S_{\ell-1}), S_\ell) \leftarrow j_\ell} \right\}.$$

Here the expression $R^{(\mu(S_1),\ldots,\mu(S_{\ell-1}),S_\ell)\leftarrow j_\ell}$ is obtained from R by inserting the index j_ℓ into $R_\ell(\mu(S_1), \ldots, \mu(S_{\ell-1}), S_\ell)$. Then

$$E_\ell = \bigcup_{(S_1,\ldots,S_{\ell-1}) \in (2^{[c_1]} \setminus \{\emptyset\}) \times \cdots \times (2^{[c_{\ell-1}]} \setminus \{\emptyset\})} E_\ell(S_1, \ldots, S_{\ell-1}).$$

Proof. Fix $S_1, \dots, S_{\ell-1}$ and suppose $(v^{\ell-1}_{R,j_\ell}, v^{\ell}_{R'}) \in E_\ell(S_1, \dots, S_{\ell-1})$. Then by Lemma 5.5.2, since S_ℓ is chosen so that $S_\ell \cap \mathcal{I}(v^{\ell-1}_{R,j_\ell}) = \emptyset$, $d^{\rightarrow}_{R'}(j_\ell) = 0$, and thus $j_\ell \in L^-(v^{\ell}_{R'})$, so $(v^{\ell-1}_{R,j_\ell}, v^{\ell}_{R'}) \in E_\ell$.

For the other direction, suppose $\left(v^{\ell-1}_{R'\setminus\{j_\ell\},j_\ell}, v^{\ell}_{R'}\right) \in E_\ell$, and let $S^*_1, \dots, S^*_\ell$ be such that $v^{\ell}_{R'} \in V_\ell(S^*_1, \dots, S^*_\ell)$. Then R' is obtained from $R' \setminus \{j_\ell\}$ by adding j_ℓ to the component $R_\ell(\mu(S^*_1), \dots, \mu(S^*_{\ell-1}), S^*_\ell)$. Then since $j_\ell \in L^-(v^{\ell}_{R'})$, $d^{\rightarrow}_{R'}(j_\ell) = 0$, so by Lemma 5.5.2, $S^*_\ell \in 2^{[c_\ell]\setminus\mathcal{I}(v^{\ell-1}_{R,j_\ell})} \setminus \{\emptyset\}$, and so $\left(v^{\ell-1}_{R'\setminus\{j_\ell\},j_\ell}, v^{\ell}_{R'}\right) \in E_\ell(S^*_1, \dots, S^*_{\ell-1})$. □

While E_ℓ represents all edges between $V^+_{\ell-1}$ and V_ℓ, we now define two sets of edges $\tilde{E}_\ell \subset E_\ell$, and $\tilde{E}'_\ell$ disjoint from $V^+_{\ell-1} \times V_\ell$, that each solve a different technical issue. First, in Section 5.5.5, we will see that the complexity of transitions in E_ℓ depends on the number of collisions between the new block $A^{(j_\ell)}_\ell$ being added, and $(\ell-1)$-collisions already stored in $D_{\ell-1}(R)$, so we will only attempt to implement the transition subroutine correctly when this set is not too large. In anticipation of this, we define:

$$\tilde{E}_\ell := \left\{(v^{\ell-1}_{R,j_\ell}, v^{\ell}_{R'}) \in E_\ell : |\mathcal{K}(\overline{R}_1, \dots, \overline{R}_{\ell-1}, A^{(j_\ell)}_\ell)| \geq p_\ell\right\} \subset E_\ell, \tag{5.58}$$

where $p_\ell \in \mathsf{polylog}(n)$, which will be part of $\tilde{E}$, the set of edges whose transitions we fail to implement. Second, if any $v^{\ell-1}_{R,j_\ell} \in V^+_{\ell-1}$ has no neighbour in V_ℓ, which happens exactly when $\mathcal{I}(v^{\ell-1}_{R,j_\ell}) = [c_\ell]$, then its correct star state would simply have one incoming edge from V_0, which, as discussed in Section 5.4.2, would make it impossible to define a flow satisfying all star state constraints (**P2** of Theorem 4.3.1). Unlike in the case of E_1, there is no constant c_ℓ such that we can assume $\mathcal{I}(v^{\ell-1}_{R,j_\ell}) \subsetneq [c_\ell]$ for all $v^{\ell-1}_{R,j_\ell} \in V^+_{\ell-1}$. That is because while each $i_\ell \in A_\ell$ can have at most $k-2$ collisions in $R_{\ell+1}$, the total number of such collisions for all $i_\ell \in A^{(j_\ell)}_\ell$ may be linear in $|A^{(j_\ell)}_\ell|$. Fortunately this happens for only a very small fraction of $v^{\ell-1}_{R,j_\ell} \in V^+_{\ell-1}$. Thus, we define (choosing $\{1\}$ arbitrarily):

$$\tilde{E}'_\ell := \left\{\left(v^{\ell-1}_{R,j_\ell}, f_{v^{\ell-1}_{R,j_\ell}}(\{1\})\right) : \mathcal{I}(v^{\ell-1}_{R,j_\ell}) = [c_\ell]\right\}, \tag{5.59}$$

which is disjoint from E_ℓ. Note that for $\ell = k-1$, we always have $\mathcal{I}(v^{k-2}_{R,j_{k-1}}) = \emptyset$ and $[c_{k-1}] = [1]$, so $\tilde{E}'_{k-1} = \emptyset$. Since $\tilde{E}'_\ell$ will be part of $\tilde{E}$, we assume its endpoints $f_{v^{\ell-1}_{R,j_\ell}}(\{1\})$ are just otherwise isolated vertices that we do not consider a part of $V(G)$ (see Remark 4.3.2). As with E_ℓ, we set all edges in $\tilde{E}'_\ell$ to have weight w_ℓ. Thus, from this discussion as well as Lemma 5.5.4 we have, for all $v^{\ell-1}_{R,j_\ell} \in V^+_{\ell-1}$ and $S_\ell \in L^+(v^{\ell-1}_{R,j_\ell})$,

$$\mathsf{w}_{v^{\ell-1}_{R,j_\ell},S_\ell} = \begin{cases} \mathsf{w}_\ell = \sqrt{n/m_\ell} & \text{if } S_\ell \cap \mathcal{I}(v^{\ell-1}_{R,j_\ell}) = \emptyset \\ \mathsf{w}_\ell = \sqrt{n/m_\ell} & \text{if } \mathcal{I}(v^{\ell-1}_{R,j_\ell}) = [c_\ell] \text{ and } S_\ell = \{1\} \\ 0 & \text{else.} \end{cases} \tag{5.60}$$

We can see from (5.43), that

$$|E_\ell| + \left|\tilde{E}'_\ell\right| \leq \left|L^+(v^{\ell-1}_{R,j_\ell})\right| \left|V^+_{\ell-1}\right| = O\left(n\,|V_0|\right). \tag{5.61}$$

$E_k \subset V_{k-1} \times V_k$: Finally, there is an edge between $v^{k-1}_R \in V_{k-1}$ and $v^k_{R,i_k} \in V_R$ for any $i_k \in A_k$, so we define

$$L^+(v^{k-1}_R) := A_k \text{ and } L^-(v^k_{R,i_k}) := \{\leftarrow\},$$

Edge Set	(u,v)	(u,i)	(v,j)	$\mathsf{w}_{u,v}$	$\mathsf{T}_{u,v}$
$E_0^+ \subset V_0 \times V_0^+$	(v_R^0, v_{R,i_1}^0)	(v_R^0, i_1)	$(v_{R,i_1}^0, \leftarrow)$	$\mathsf{w}_0^+ = 1$	$\mathsf{T}_0^+ = \widetilde{O}(1)$
$E_1 \subset V_0^+ \times V_1$	$(v_{R,i_1}^0, v_{R'}^1)$	(v_{R,i_1}^0, S_1)	$(v_{R'}^1, i_1)$	$\mathsf{w}_1 = 1$	$\mathsf{T}_1 = \widetilde{O}(1)$
$\{E_\ell^+ \subset V_\ell \times V_\ell^+\}_{\ell=1}^{k-2}$	$(v_R^1, v_{R,j_{\ell+1}}^1)$	$(v_R^1, j_{\ell+1})$	$(v_{R,j_{\ell+1}}^1, \leftarrow)$	$\mathsf{w}_\ell^+ = 1$	$\mathsf{T}_\ell^+ = \widetilde{O}(1)$
$\{E_\ell \subset V_{\ell-1}^+ \times V_\ell\}_{\ell=1}^{k-1}$	$(v_{R,j_\ell}^{\ell-1}, v_{R'}^\ell)$	$(v_{R,j_\ell}^{\ell-1}, S_\ell)$	$(v_{R'}^\ell, j_\ell)$	$\mathsf{w}_\ell = \sqrt{\frac{n}{m_\ell}}$	$\mathsf{T}_\ell = \widetilde{O}\left(\sqrt{\frac{n}{m_\ell}}\right)$
$E_k \subset V_{k-1}^+ \times V_k$	(v_R^{k-1}, v_{R,i_k}^k)	(v_R^{k-1}, i_k)	$(v_{R,i_k}^k, \leftarrow)$	$\mathsf{w}_k = 1$	$\mathsf{T}_k = \widetilde{O}(1)$

Table 5.5: For each edge in $\overrightarrow{E}(G)$, we can describe it in three ways: as a pair of vertices (u,v); as a vertex u and forward label $i = f_u^{-1}(v)$; and as a vertex v and backward label $j = f_v^{-1}(u)$ (see Definition 3.2.13). We summarise these three descriptions for the edge sets that make up $\overrightarrow{E}(G) \setminus \tilde{E}$, along with the edge weights, and transitions costs (see Corollary 5.5.14). The edge labels i and j range across (sometimes strict) subsets of $L^+(u)$ and $L^-(u)$ (see Table 5.4). For example, for $u \in V_0^+$, $i = S_1 \in L^+(u) = 2^{[c_1]} \setminus \{\emptyset\}$, (u,i) only represents an edge of $\overrightarrow{E}(G)$ when $S_1 \cap \mathcal{I}(u) = \emptyset$ (see Lemma 5.5.3 and (5.60)).

and let $f_{v_R^{k-1}}(i_k) = v_{R,i_k}^k$, and $f_{v_{R,i_k}^k}(\leftarrow) = v_R^{k-1}$. We let E_k be the set of such edges:

$$E_k := \left\{(v_R^{k-1}, v_{R,i_k}^k) : v_R^{k-1} \in V_{k-1}, i_k \in A_k\right\},$$

and we set $\mathsf{w}_e = \mathsf{w}_k = 1$ for all $e \in E_k$. This, together with (5.48), implies that

$$|E_k| = O\left(\frac{n^2}{t_{k-1}}|V_0|\right). \tag{5.62}$$

The graph G: The full graph G is defined by

$$V(G) = \bigcup_{\ell=0}^{k} V_\ell \cup \bigcup_{\ell=0}^{k-2} V_\ell^+$$

$$\overrightarrow{E}(G) = \{(u,v) : u \in V(G), i \in L^+(u) : \mathsf{w}_{u,i} \neq 0\} = \bigcup_{\ell=0}^{k-2} E_\ell^+ \cup E_1 \cup \bigcup_{\ell=2}^{k-1} (E_\ell \cup \tilde{E}_\ell') \cup E_k,$$

where the edge label sets $L^+(u)$ are summarised in Table 5.4, and weights are summarised in Table 5.5. We define (recall that $\tilde{E}_\ell \subset E_\ell$):

$$\tilde{E} := \bigcup_{\ell=2}^{k-1} (\tilde{E}_\ell \cup \tilde{E}_\ell'). \tag{5.63}$$

The marked set and checking cost: In the notation of Theorem 4.3.1, we let $V_{\mathsf{M}} = V_k$, and we define a subset $M \subseteq V_{\mathsf{M}}$ as follows. If $(a_1, \ldots, a_k) \in A_1 \times \cdots \times A_k$ is the unique k-collision, we let

$$\begin{aligned} M &= \left\{v_{R_1,\ldots,R_{k-1},i_k}^k \in V_k : \exists (i_1, \ldots, i_{k-1}, x_{i_1}) \in D_{k-1}(R) \text{ s.t. } x_{i_1} = x_{i_k}\right\} \\ &= \left\{v_{R_1,\ldots,R_{k-1},i_k}^k : \exists S_1 \subseteq [c_1], \ldots, S_{k-1} \subseteq [c_{k-1}], s_1 \in S_1, \ldots, s_{k-1} \in S_{k-1} \text{ s.t.} \right. \\ &\qquad \left. \forall \ell \in [k-1],\ a_\ell \in \overline{R}_\ell(s_1, \ldots, s_{\ell-1}, S_\ell) \text{ and } i_k = a_k\right\}, \end{aligned} \tag{5.64}$$

and otherwise, if there is no k-collision, $M = \emptyset$. We can decide whether $v^k_{R,i_k} \in V_k$ is marked by querying i_k to obtain the value x_{i_k} and looking it up in $D_{k-1}(R)$ to see if we find some $(i_1, \ldots, i_{k-1}, x_{i_k})$, in which case, it must be that $a_1 = i_1, \ldots, a_k = i_k$. Thus, the checking cost is at most

$$\mathsf{C} = O(\log n). \tag{5.65}$$

5.5.3 The star states and their generation

We define the set of alternative neighbourhoods (Definition 4.2.1) with which we will apply Theorem 4.3.1. For $\ell \in [k]_0$, for all $v^\ell_R \in V_\ell$, we add a single star state to $\Psi_\star(u)$, which has one of three forms, depending on ℓ (refer to Table 5.4): for $v^0_R \in V_0$,

$$|\psi^G_\star(v^0_R)\rangle = \sum_{i_1 \in A_1 \setminus R_1} \sqrt{\mathsf{w}^+_0}\, |v^0_R, i_1\rangle; \tag{5.66}$$

for $\ell \in [k-1]$, and $v^\ell_R \in V_\ell(S_1, \ldots, S_\ell)$,

$$|\psi^G_\star(v^\ell_R)\rangle = -\sum_{\substack{j_\ell \in R_\ell(\mu(S_1),\ldots,\mu(S_{\ell-1}),S_\ell):\\ d^{\rightarrow}_R(j_\ell)=0}} \sqrt{\mathsf{w}_\ell}\, |v^\ell_R, \leftarrow, j_\ell\rangle + \sum_{j_{\ell+1} \in [m_{\ell+1}] \setminus R_{\ell+1}} \sqrt{\mathsf{w}^+_\ell}\, |v^\ell_R, \rightarrow, j_{\ell+1}\rangle; \tag{5.67}$$

and finally, for $v^k_{R,i_k} \in V_k$,

$$|\psi^G_\star(v^k_{R,i_k})\rangle = -\sqrt{\mathsf{w}_k}\, |v^k_{R,i_k}, \leftarrow\rangle. \tag{5.68}$$

Here we have explicitly included the $\rightarrow$ and $\leftarrow$ parts of each element of L^+ and L^-, which are normally left implicit, in order to stress that the two sum are orthogonal. From Table 5.4, along with the description of w in Lemma 5.5.3, we can see that for $v^0_{R,i_1} \in V^+_0$,

$$|\psi^G_\star(v^0_{R,i_1})\rangle = -\sqrt{\mathsf{w}^+_0}\, |v^0_{R,i_1}, \leftarrow\rangle + \sum_{S_1 \in 2^{[c_1] \setminus \mathcal{I}(v^0_{R,i_1})} \setminus \{\emptyset\}} \sqrt{\mathsf{w}_1}\, |v^0_{R,i_1}, S_1\rangle.$$

To generate this state, one would have to compute $\mathcal{I}(v^0_{R,i_1})$ (see (5.52)), which would require determining the locations of all forward collisions of i_1, which is far too expensive. Hence, we simply add all options to $\Psi_\star(v^0_{R,i_1})$ (we see in Lemma 5.5.5 that generating this *set* is not difficult):

$$\Psi_\star(v^0_{R,i_1}) := \bigcup_{\mathcal{I}_1 \subsetneq [c_1]} \left\{ |\psi^{\mathcal{I}_1}_\star(v^0_{R,i_1})\rangle := -\sqrt{\mathsf{w}^+_0}\, |v^0_{R,i_1}, \leftarrow\rangle + \sum_{S_1 \in 2^{[c_1] \setminus \mathcal{I}_1} \setminus \{\emptyset\}} \sqrt{\mathsf{w}_1}\, |v^0_{R,i_1}, S_1\rangle \right\}. \tag{5.69}$$

Thus, since we always have $\mathcal{I}(v^0_{R,i_1}) \subsetneq [c_1]$, $|\psi^G_\star(v^0_{R,i_1})\rangle = |\psi^{\mathcal{I}(v^0_{R,i_1})}_\star(v^0_{R,i_1})\rangle \in \Psi_\star(v^0_{R,i_1})$. Similarly, for $\ell \in \{2, \ldots, k-1\}$ and $v^{\ell-1}_{R,j_\ell} \in V^+_{\ell-1}$ define:

$$\Psi_\star(v^{\ell-1}_{R,j_\ell}) := \bigcup_{\mathcal{I}_\ell \subsetneq [c_\ell]} \left\{ |\psi^{\mathcal{I}_\ell}_\star(v^{\ell-1}_{R,j_\ell})\rangle := -\sqrt{\mathsf{w}^+_{\ell-1}}\, |v^{\ell-1}_{R,j_\ell}, \leftarrow\rangle + \sum_{S_\ell \in 2^{[c_\ell] \setminus \mathcal{I}_\ell} \setminus \{\emptyset\}} \sqrt{\mathsf{w}_\ell}\, |v^{\ell-1}_{R,j_\ell}, S_\ell\rangle \right\}. \tag{5.70}$$

Then from (5.60), we have

$$|\psi^G_\star(v^\ell_{R,j_\ell})\rangle = \begin{cases} |\psi^{\mathcal{I}(v^\ell_{R,j_\ell})}_\star(v^\ell_{R,j_\ell})\rangle & \text{if } \mathcal{I}(v^\ell_{R,j_\ell}) \subsetneq [c_\ell] \\ |\psi^{[c_\ell]\setminus\{1\}}_\star(v^\ell_{R,j_\ell})\rangle & \text{if } \mathcal{I}(v^\ell_{R,j_\ell}) = [c_\ell], \end{cases} \tag{5.71}$$

where $\mathcal{I}(v^{\ell}_{R,j_\ell})$ is defined in (5.53).

We now describe how to generate the states in $\bigcup_{u\in V(G)} \Psi_\star(u)$ in $\widetilde{O}(1)$ complexity (see Definition 4.2.1):

Lemma 5.5.5. *The states $\Psi_\star = \{\Psi_\star(u)\}_{u\in V(G)}$ can be generated in $\widetilde{O}(1)$ complexity.*

Proof. The description of a vertex $u \in V(G)$ begins with a label indicating to which of $V_0,\dots,V_k$ or $V_0^+,\dots,V_{k-2}^+$ it belongs, so we can define subroutines $U_0,\dots,U_k$ and $U_{0,+},\dots,U_{k-2,+}$ that generate the star states in each vertex set respectively, and then

$$U_\star = \sum_{\ell=0}^{k} |\ell\rangle\langle\ell| \otimes U_\ell + \sum_{\ell=0}^{k-2} |\ell,+\rangle\langle\ell,+| \otimes U_{\ell,+}$$

will generate the star states in the sense of Definition 4.2.1.

We begin with U_0. For $v_R^0 \in V_0$, we have $\Psi_\star(v_R^0) = \{|\psi_\star^G(v_R^0)\rangle\}$, where $|\psi_\star^G(v_R^0)\rangle$ is as in (5.66). Thus, implementing the map $U_0 : |u\rangle|0\rangle \mapsto\propto |\psi_\star^G(u)\rangle$ is as simple as generating a uniform superposition over A_1, and then using $O(\log n)$ rounds of amplitude amplification to get inverse polynomially close to the uniform superposition over $A_1 \setminus R_1$.

For $\ell \in [k-1]$, and $v_R^\ell \in V_\ell$, we again have $\Psi_\star(v_R^\ell) = \{|\psi_\star^G(v_R^\ell)\rangle\}$, where $|\psi_\star^G(v_R^\ell)\rangle$ is as in (5.67). To implement $U_\ell : |u\rangle|0\rangle \mapsto\propto |\psi_\star^G(u)\rangle$, we first compute (referring to Table 5.5 for the weights):

$$|u,0\rangle \mapsto\propto |u\rangle \left(-\sqrt{\mathsf{w}_\ell}|\leftarrow\rangle + \sqrt{\mathsf{w}_\ell^+}|\rightarrow\rangle\right)|0\rangle = |u\rangle \left(-(n/m_\ell)^{1/4}|\leftarrow\rangle + |\rightarrow\rangle\right)|0\rangle,$$

which can be implemented by a $O(1)$-qubit rotation. Then conditioned on $\leftarrow$, generate a uniform superposition over $j_\ell \in R_\ell(\mu(S_1^*),\dots,\mu(S_{\ell-1}^*),S_\ell^*)$ (we can learn the sets $S_1^*,\dots,S_\ell^*$ by seeing which sets are bigger, or assume we simply keep track of these values in some convenient way), and then using $O(\log n)$ rounds of amplitude amplification to get inverse polynomially close to the superposition over such j_ℓ such that $d_R^{\rightarrow}(j_\ell) = 0$, which we can check by looking up j_ℓ in $C_\ell^{\rightarrow}(R)$. We have used the fact that our data structure supports taking a uniform superposition (see Section 3.5.1). Finally, conditioned on $\rightarrow$, generate a uniform superposition over $j_{\ell+1} \in [m_{\ell+1}]$, and use $O(\log n)$ rounds of amplitude amplification to get inverse polynomially close to a superposition over $j_{\ell+1} \in [m_{\ell+1}] \setminus R_{\ell+1}$.

For $\ell \in \{2,\dots,k-1\}$ (the case for $\ell = 0$ is nearly identical), and $v^{\ell-1}_{R,j_\ell} \in V^+_{\ell-1}$, $\Psi_\star(v^{\ell-1}_{R,j_\ell})$ is a set of multiple states, as in (5.70). To implement $U_{\ell-1,+} : |v^{\ell-1}_{R,j_\ell}\rangle|\mathcal{I}_\ell\rangle \mapsto |\psi_\star^{\mathcal{I}_\ell}(v^{\ell-1}_{R,j_\ell})\rangle$ for all $\mathcal{I}_\ell \subsetneq [c_\ell]$, we note that each of these states is just $|v^{\ell-1}_{R,j_\ell}\rangle$ tensored with a constant-sized state depending only on $\mathcal{I}_\ell$, so we can implement $U_{\ell-1,+}$ in $O(1)$ time, by Lemma 5.4.3.

Finally, we implement U_k. For $v^k_{R,i_k} \in V_k$, we have $\Psi_\star(v^k_{R,i_k}) = \{|\psi_\star^G(v^k_{R,i_k})\rangle\}$ as in (5.68), so implementing the map $U_k : |v^k_{R,i_k}\rangle|0\rangle \mapsto\propto |\psi_\star^G(v^k_{R,i_k})\rangle \propto |v^k_{R,i_k}\rangle|\leftarrow\rangle$ is trivial. We thus conclude that $U_\star$ can be implemented in $\mathsf{polylog}(n) = \widetilde{O}(1)$ complexity. □

5.5.4 Tail bounds on number of collisions

If $\overline{R}_1,\dots,\overline{R}_{k-1}$ were uniform random subsets of $A_1,\dots,A_{k-1}$ respectively, it would be simple to argue that, for example, the number of collisions stored in $D_\ell(R)$ for any ℓ, which is a subset of $\mathcal{K}(\overline{R}_1,\dots,\overline{R}_\ell)$, is within a constant of the average, with high probability. Since $\overline{R}_\ell$ is instead chosen from A_ℓ by taking t_ℓ blocks of A_ℓ, and these blocks themselves are not uniform random, but rather chosen by a d-wise independent permutation for some $d = \mathsf{polylog}(n)$, proving the necessary bounds, which are needed to upper bound the setup and transitions costs, is somewhat more subtle.

Lemma 5.5.6. *For any $\ell', \ell \in [k-1]$ where $\ell > \ell'$ and for any constant κ there exists a constant c such that the following holds. If $v_R^{\ell-1}$ is chosen uniformly at random from $V_{\ell-1}$, then for any fixed (non-random) $j \in [m_\ell]$ we have*

$$\Pr\left[\left|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell'}, A_\ell^{(j)})\right| \geq c\frac{t_{\ell'}}{m_\ell}\log^{2^{\ell'-1}}(n)\right] \leq n^{-\kappa}.$$

Proof. The proof proceeds by induction on ℓ'.
Base case: For $\ell' = 1$, we have $\overline{R}_1 = R_1$ is a uniform random subset of A_1 of size $\Theta(t_1)$ (for this proof, we ignore the partition of R_1 into $(R_1(S_1))_{S_1}$). Fix $A_\ell^{(j)}$ for some $\ell > 1$. Then $Z = |\mathcal{K}(\overline{R}_1, A_\ell^{(j)})|$ is a hypergeometric random variable, where we draw R_1 from A_1 and where we consider any $i_1 \in R_1$ marked whenever it is part of a collision in $\mathcal{K}(\overline{R}_1, A_\ell^{(j)})$. This means Z has parameters $N = |A_1| = \Theta(n)$, $K = |\mathcal{K}(A_1, A_\ell^{(j)})|$ and $d = |R_1| = \Theta(t_1)$. As mentioned in Section 5.3, we may assume for any any $A_\ell^{(j)}$ that $K = \Theta\left(\left|A_\ell^{(j)}\right|\right) = \Theta(n/m_\ell)$. Then for any constant c, we have $c(t_1/m_\ell)\log n \geq 7Kd/N$ for sufficiently large n, so by Lemma 5.2.1:

$$\Pr\left[\left|\mathcal{K}(\overline{R}_1, A_\ell^{(j)})\right| \geq c\frac{t_1}{m_\ell}\log n\right] \leq e^{-c\frac{t_1}{m_\ell}\log n} = n^{-ct_1/m_\ell}.$$

Referring to Table 5.3, we have

$$\frac{t_1}{m_\ell} = \Theta\left(\frac{t_1}{t_{\ell-1}}\right) \geq \Omega\left(\frac{t_1}{t_1}\right) = \Omega(1).$$

Hence, we can choose c sufficiently large so that $n^{-ct_1/m_\ell} \leq n^{-\kappa}$, completing the base case.
Induction step: Suppose that for some $\ell' - 1 \in [k-2]$ and all $\ell \in \{\ell', \ldots, k-1\}$ the lemma holds; i.e. for any fixed $j \in [m_\ell]$ and for any κ' there exists a constant c' such that

$$\Pr\left[\left|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell'-1}, A_\ell^{(j)})\right| \geq c'\frac{t_{\ell'-1}}{m_\ell}\log^{2^{\ell'-2}}(n)\right] \leq n^{-\kappa'},$$

where R_1 is a uniform random subset of A_1 of size $\Theta(t_1)$, and for all $\ell'' \in \{2, \ldots, \ell'-1\}$, $R_{\ell''}$ is a uniform random subset of $[m_{\ell''}]$ of size $\Theta(t_{\ell''})$.

Now consider $Z = |\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell'}, A_\ell^{(j)})|$, where $\ell > \ell'$ and where $R_{\ell'}$ is a uniform random subset of $[m_{\ell'}]$ of size $\Theta(r_{\ell'})$, and $A_\ell^{(j)}$ is still fixed. It is important to remark that this does not imply that $\overline{R}_{\ell'}$ is uniformly random, so instead we look at the blocks of $A_{\ell'}$. We say that any block $A_{\ell'}^{(j')}$ of $A_{\ell'}$ is *marked* if it collides with an ℓ'-collision in $\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell'-1}, A_\ell^{(j)})$, and we let B_1 be the random variable that counts the number of such marked blocks in $A_{\ell'}$. Let $\mathcal{E}_1$ be the event that $|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell'-1}, A_\ell^{(j)})| < c'\frac{t_{\ell'-1}}{m_\ell}\log^{2^{\ell'-2}}$, which happens with probability at least $1 - n^{-\kappa'}$, by the induction hypothesis. Assuming $\mathcal{E}_1$ directly implies an upper bound on B_1 of the form

$$B_1 \leq c'\frac{t_{\ell'-1}}{m_\ell}\log^{2^{\ell'-2}}(n). \tag{5.72}$$

Next we introduce a hypergeometric random variable B_2 that counts the number of marked blocks of $R_{\ell'}$, which we draw uniformly from $\{A_{\ell'}^{(j')}\}_{j' \in [m_{\ell'}]}$, which has B_1 marked blocks. Conditioned on event $\mathcal{E}_1$, this means that B_2 has parameters $N = m_{\ell'}$, $K = B_1 \leq c'\frac{t_{\ell'-1}}{m_\ell}\log^{2^{\ell'-2}}(n)$ (see (5.72)) and $d = |R_{\ell'}| = \Theta(t_{\ell'})$. To relate $Z = |\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell'}, A_\ell^{(j)})|$ to B_2, we need to analyse what the effect is of any marked block in $R_{\ell'}$ on the number

of collisions in $|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell'}, A_\ell^{(j)})|$. By the induction hypothesis we know that for any block $A_{\ell'}^{(j')}$ of $A_{\ell'}$ and each κ'', there exists a constant c'' such that:

$$\Pr\left[\left|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell'-1}, A_{\ell'}^{(j')})\right| \geq c'' \log^{2^{\ell'-2}}(n)\right] \leq n^{-\kappa''}.$$

If $|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell'-1}, A_{\ell'}^{(j')})| < c'' \log^{2^{\ell'-2}}(n)$ for every block $A_{\ell'}^{(j')}$ of $A_{\ell'}$, which we denote by event $\mathcal{E}_2$, then any marked block that gets added to $R_{\ell'}$ results in at most $c'' \log^{2^{\ell'-2}}(n)$ collisions in $\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell'}, A_\ell^{(j)})$, so $Z \leq B_2 c'' \log^{2^{\ell'-2}}(n)$. This implies that

$$\Pr\left[Z \geq cc'' \frac{t_{\ell'}}{m_\ell} \log^{2^{\ell'-1}}(n) \middle| \mathcal{E}_1 \wedge \mathcal{E}_2\right] \leq \Pr\left[B_2 \geq c \frac{t_{\ell'}}{m_\ell} \log^{2^{\ell'-1}-2^{\ell'-2}} n \middle| \mathcal{E}_1\right]. \tag{5.73}$$

Since B_2 is a hypergeometric random variable, and $c\frac{t_{\ell'}}{m_\ell} \log^{2^{\ell'-1}-2^{\ell'-2}} n \geq 7dK/N$ for sufficiently large c, we can use Lemma 5.2.1 and $m_{\ell'} = \Theta(t_{\ell'-1})$ to derive:

$$\Pr\left[B_2 \geq c \frac{t_{\ell'}}{m_\ell} \log^{2^{\ell'-1}-2^{\ell'-2}} n \middle| \mathcal{E}_1\right] \leq e^{-c\frac{t_{\ell'}}{m_\ell} \log^{2^{\ell'-1}-2^{\ell'-2}} n} \leq n^{-ct_{\ell'}/m_\ell}, \tag{5.74}$$

since $2^{\ell'-1} - 2^{\ell'-2} \geq 1$ whenever $\ell' \geq 2$. By Table 5.3, we have

$$\frac{t_{\ell'}}{m_\ell} = \Theta\left(\frac{t_{\ell'}}{t_{\ell-1}}\right) \geq \Omega\left(\frac{t_{\ell'}}{t_{\ell'}}\right) = \Omega(1).$$

Hence, by (5.73) and (5.74), we can choose c sufficiently large such that:

$$\Pr\left[Z \geq cc'' \frac{t_{\ell'}}{m_\ell} \log^{2^{\ell'-1}}(n) \middle| \mathcal{E}_1 \wedge \mathcal{E}_2\right] \leq n^{-ct_{\ell'}/m_\ell} \leq n^{-\kappa'}. \tag{5.75}$$

The only thing left to do is to use the union bound to upper bound

$$\Pr\left[\neg\left(\mathcal{E}_1 \wedge \mathcal{E}_2\right)\right] \leq n^{-\kappa'} + m_{\ell'} n^{-\kappa''} \leq n^{-\kappa'} + n^{-\kappa''+1}, \tag{5.76}$$

where in the final inequality we have used the assumption from the lemma that $m_{\ell'} \leq n$. We can now combine (5.75) and (5.76), to conclude that for any κ we can choose $\kappa' > \kappa$ and $\kappa'' > \kappa + 1$ and a constant $c_{\ell'}$ large enough such that:

$$\begin{aligned}\Pr\left[Z \geq c_\ell \frac{t_{\ell'}}{m_\ell} \log^{2^{\ell'-1}}(n)\right] &\leq \Pr\left[Z \geq c_\ell \frac{t_{\ell'}}{m_\ell} \log^{2^{\ell'-1}}(n) \middle| \mathcal{E}_1 \wedge \mathcal{E}_2\right] + \Pr\left[\neg\left(\mathcal{E}_1 \wedge \mathcal{E}_2\right)\right] \\ &\leq n^{-\kappa'} + n^{-\kappa'} + n^{-\kappa''+1} \leq n^{-\kappa}.\end{aligned}$$

□

Corollary 5.5.7. *For $\ell \in \{2, \ldots, k-2\}$, let $v_R^{\ell-1}$ be chosen uniformly at random from $V_{\ell-1}$. If the partition of $A_{\ell+1}$ into $\bigcup_{j \in [m_{\ell+1}]} A_{\ell+1}^{(j)}$ is chosen by a d-wise independent permutation for $d = \log^{2^{k-1}}(n)$ (see Section 5.3), then for all $j \in [m_\ell]$,*

$$\Pr\left[|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell-1}, A_\ell^{(j)}, \overline{R}_{\ell+1})| \geq 1\right] \leq o(1),$$

and for any constant κ there exists a constant c such that:

$$\Pr\left[|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell-1}, A_\ell^{(j)}, \overline{R}_{\ell+1})| \geq c\right] \leq n^{-\kappa},$$

where we note that $|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell-1}, A_\ell^{(j)}, \overline{R}_{\ell+1})|$ is an upper bound on the number of potential faults if we add the block $A_\ell^{(j)}$ to $\overline{R}_\ell$.

Proof. By Lemma 5.5.6, for any κ' there exists a c' large enough such that for each fixed $j \in [m_\ell]$ we have

$$\Pr\left[|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell-1}, A_\ell^{(j)})| \geq c' \log^{2^{\ell-1}}(n)\right] \leq n^{-\kappa'}.$$

Let $i^1, \ldots, i^K \in A_{\ell+1}$ be all the points that collide with $\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell-1}, A_\ell^{(j)})$, so with high probability there are

$$K < c'k \log^{2^{\ell-1}}(n) \leq \log^{2^{k-1}}(n) = d$$

of them, since each tuple in $\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell-1}, A_\ell^{(j)})$ can collide with less than k other indices. Let $\tau : [n] \to [n]$ be the d-wise independent permutation used to choose the partitions of $[n]$, as in (5.1). Then $\{\tau(i^1), \ldots, \tau(i^K)\}$ is distributed as uniform set of size K, as long as $K \leq d$. In that case, the number of elements of $\{i^1, \ldots, i^K\}$ that are included in $\overline{R}_{\ell+1}$, Z, is a hypergeometric random variable with $N = |A_{\ell+1}| = \Theta(n)$, $K < c'k \log^{2^{\ell-1}}(n)$ as above, and $d = |\overline{R}_{\ell+1}| = \Theta\left(\frac{nt_{\ell+1}}{m_{\ell+1}}\right)$ draws. From Table 5.3, we have

$$\frac{dK}{N} = \Theta\left(\frac{t_{\ell+1}}{m_{\ell+1}} \log^{2^{\ell-1}}(n)\right) = \Theta\left(n^{-\frac{2^{k-\ell-2}}{2^{k-1}}} \log^{2^\ell - 1}(n)\right) \leq n^{-\Omega(1)}.$$

Thus, there is some constant ϵ such that $dK/N \leq n^{-\epsilon}$, so for any constant c, we can use the union bound and Corollary 5.2.2 to get

$$\Pr[Z \geq c] \leq 2e^c \left(cn^\epsilon\right)^{-c} = 2\left(\frac{e}{c}\right)^c n^{-\epsilon c} + n^{-\kappa'}.$$

Hence, by choosing $\kappa' > \kappa$ and c large enough we obtain $\Pr[Z \geq c] \leq n^{-\kappa}$. Moreover, we also see that for $c = 1$:

$$\Pr[Z \geq 1] \leq O(n^{-\epsilon} + n^{-\kappa'}) \leq o(1). \qquad \square$$

Corollary 5.5.8. *Let v_R^ℓ be chosen uniformly at random from V_ℓ. For any constant κ, there exist a constant c such that:*

$$\Pr\left[|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_\ell)| \geq ct_\ell \log^{2^{\ell-2}}(n)\right] \leq n^{-\kappa}.$$

Proof. By Lemma 5.5.6, for each fixed $j \in [m_\ell]$ and constant $\kappa' > \kappa + 1$ there exists a c large enough such that

$$\Pr\left[|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell-1}, A_\ell^{(j)})| \geq c \log^{2^{\ell-2}}(n)\right] \leq n^{-\kappa'}.$$

By the union bound, the probability of this bad event happening for *any* $j \in R_\ell$ is at most $t_\ell n^{-\kappa'} \leq n^\kappa$, since $t_\ell < n$, from which the statement follows. $\square$

5.5.5 The transition subroutines

In this section we show how to implement the transition map $|u, i\rangle \mapsto |v, j\rangle$ for $(u, v) \in \overrightarrow{E}(G)$ with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$ (see Definition 3.2.13). We do this by exhibiting uniform (see Lemma 3.5.2) subroutines $\mathcal{S}_1, \ldots, \mathcal{S}_k, \mathcal{S}_{0,+}, \ldots, \mathcal{S}_{k-2,+}$ that implement the transitions in each of the edge sets E_1, ...,E_k, E_0^+, ..., E_{k-2}^+ defined in Section 5.5.2, whose union is $\overrightarrow{E}(G) \setminus \tilde{E}$. In Corollary 5.5.14, we will combine these to get a quantum subroutine (Definition 3.5.1) for the full transition map.

Lemma 5.5.9. *For $\ell \in [k-2]_0$, there is a subroutine $\mathcal{S}_{\ell,+}$ such that for all $(u, v) \in E_\ell^+$ with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$, $\mathcal{S}_{\ell,+}$ maps $|u, i\rangle$ to $|v, j\rangle$ with error 0 in complexity $\mathsf{T}_{u,v} = \mathsf{T}_\ell^+ = \widetilde{O}(1)$.*

Proof. The proof is identical to that of Lemma 5.4.5. □

Lemma 5.5.10. *There is a uniform subroutine $\mathcal{S}_1$ such that for all $(u,v) \in E_1$ with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$, $\mathcal{S}_1$ maps $|u,i\rangle$ to $|v,j\rangle$ with error 0 in complexity $\mathsf{T}_{u,v} = \mathsf{T}_1 = \widetilde{O}(1)$.*

Proof. The proof is identical to that of Lemma 5.4.6. □

We now move on to $\mathcal{S}_\ell$, for $\ell \in \{2,\dots,k-1\}$, which is somewhat more complicated. For $(v_{R,j_\ell}^{\ell-1}, v_{R'}^{\ell}) \in E_\ell$, where $v_{R,j_\ell}^{\ell-1} \in V_{\ell-1}^{+}(S_1^*,\dots,S_{\ell-1}^*)$, meaning that R' is obtained from R by inserting j_ℓ into $R_\ell(\mu(S_1^*),\dots,\mu(S_{\ell-1}^*),S_\ell)$ for some S_ℓ, $\mathcal{S}_\ell$ should act as

$$\begin{aligned} |v_{R,j_\ell}^{\ell-1}, S_\ell\rangle &\mapsto |v_{R'}^{\ell}, j_\ell\rangle \\ \equiv |((\ell-1,+),R,D(R),j_\ell),S_\ell\rangle &\mapsto |(\ell,R',D(R')),j_\ell\rangle. \end{aligned} \tag{5.77}$$

The complexity of this map depends on $|\mathcal{K}(\overline{R}_1,\dots,\overline{R}_{\ell-1},A_\ell^{(j_\ell)})|$, which is less than $p_\ell \in \mathsf{polylog}(n)$ whenever $(v_{R,j_\ell}^{\ell-1}, v_{R'}^{\ell}) \in E_\ell \setminus \tilde{E}_\ell = E_\ell \setminus \tilde{E}$ (see (5.58) and (5.63)). Lemma 5.5.11 below describes how to implement this transition map, up to some error, in that case. For the case when $(v_{R,j_\ell}^{\ell-1}, v_{R'}^{\ell}) \in \tilde{E}_\ell \subset \tilde{E}$ we let the algorithm fail.

Lemma 5.5.11. *Fix any constant κ. For each $\ell \in \{2,\dots,k-1\}$, there is a uniform subroutine $\mathcal{S}_\ell$ that implements the transition map that maps $|u,i\rangle$ to $|v,j\rangle$ for all $(u,v) \in E_\ell \setminus \tilde{E}$ with error $O(n^{-\kappa})$, in complexity $\mathsf{T}_{u,v} = \mathsf{T}_\ell = \widetilde{O}(\sqrt{n/m_\ell})$.*

Proof. Suppose $u = v_{R,j_\ell}^{\ell} \in V_{\ell-1}^{+}(S_1^*,\dots,S_{\ell-1}^*)$. We can compute the values $S_1^*,\dots,S_{\ell-1}^*$ by checking which sets are larger, or just keeping track of these values in some convenient way, as they are chosen. Then to implement the map in (5.77), we need to insert j_ℓ into $R_\ell(\mu(S_1^*),\dots,\mu(S_{\ell-1}^*),S_\ell)$ to obtain R'_ℓ, update $D(R)$ to obtain $D(R')$, uncompute S_ℓ by checking which part of R_ℓ has size $t_\ell+1$, and increment the first register by mapping $|\ell-1,+\rangle \mapsto |\ell\rangle$. All of these take $\mathsf{polylog}(n)$ complexity, except for updating $D(R)$, which we now describe.

By (5.41), $D(R)$ consists of sets $\{D_{\ell'}(R)\}_{\ell'=1}^{k-1}$ where each $D_{\ell'}(R)$ contains a subset of $\mathcal{K}(\overline{R}_1,\dots,\overline{R}_{\ell'})$ (see (5.36)). When we go from R to R', we need to update each of these to account for any collisions involving indices $i_\ell \in A_\ell^{(j_\ell)}$ that should be recorded in $D_{\ell'}(R')$. For $\ell' < \ell$, we can see that $D_{\ell'}(R) = D_{\ell'}(R')$, since $D_{\ell'}$ only depends on $R_1,\dots,R_{\ell'}$, which are unchanged. For $\ell' > \ell$, the existence of any $(i_1,\dots,i_{\ell-1},i_\ell,i_{\ell+1},\dots,i_{\ell'},x_{i_1}) \in D_{\ell'}(R')$ (which we would now need to find and add) implies $d_R^{\rightarrow}(j_\ell) > 0$ (see (5.39)), and this is not true for any $(v_{R,j_\ell}^{\ell-1}, v_{R'}^{\ell}) \in E_\ell$ (see (5.57)). Thus, we only need to find any tuples $(i_1,\dots,i_\ell,x_{i_1})$ such that $i_\ell \in A_\ell^{(j_\ell)}$ that belong in $D_\ell(R')$. By (5.36), such a tuple should be added to $D_\ell(R')$ if and only if $(i_1,\dots,i_{\ell-1},x_{i_1}) \in D_{\ell-1}(R_{\ell-1}(\mu(S_1^*),\dots,\mu(S_{\ell-2}^*),S_{\ell-1}))$ such that $x_{i_\ell} = x_{i_1}$ for some $S_{\ell-1}$ such that $\mu(S_{\ell-1}^*) \in S_{\ell-1}$.

We search for values $i_\ell \in A_\ell^{(j_\ell)}$ such that if we look up the entry x_{i_ℓ} in the data $D_{\ell-1}(R_{\ell-1}(\mu(S_1^*),\dots,\mu(S_{\ell-2}^*),S_{\ell-1}))$ for some $S_{\ell-1}$ containing $\mu(S_{\ell-1}^*)$, we get back a non-empty set of values $(i_1,\dots,i_{\ell-1},x_{i_\ell})$. For any such value found, we add $(i_1,\dots,i_\ell,x_{i_\ell})$ to $D_\ell(R_\ell(\mu(S_1^*),\dots,\mu(S_{\ell-1}^*),S_{\ell-1}))$. This increments the value of $d_{R'}^{\rightarrow}(i_{\ell-1})$, and so if $j_{\ell-1} \in R_{\ell-1}$ is such that $i_{\ell-1} \in A_{\ell-1}^{(j_{\ell-1})}$ (we can compute $j_{\ell-1}$ from $i_{\ell-1}$ in $\widetilde{O}(1)$, see Section 5.3), we have incremented the forward collision degree of $j_{\ell-1}$, $d_{R'}^{\rightarrow}(j_{\ell-1})$. We must therefore update the entry in $C_{\ell-1}^{\rightarrow}$ for $j_{\ell-1}$. We look up $j_{\ell-1}$, and if nothing is returned, insert $(j_{\ell-1},0)$. If $(j_{\ell-1},N)$ is returned, remove it and insert $(j_{\ell-1},N+1)$. We repeat this quantum search procedure, which is uniform, until we find $p_\ell = \mathsf{polylog}(n)$ values i_ℓ, or no new i_ℓ is returned for $\kappa \log n$ times. Since we are assuming that the number of such collisions is less than p_ℓ, since $(u,v) \in E_\ell \setminus \tilde{E}_\ell$, this finds all collisions with error $O(n^{-\kappa})$, in complexity $\widetilde{O}\left(\sqrt{|A_\ell^{(j_\ell)}|}\right) = \widetilde{O}(\sqrt{n/m_\ell})$. □

We have the following corollary of the results in Section 5.5.4:

Corollary 5.5.12. *For any constant κ, there exists a choice of constants $\{c_\ell\}_{\ell=2}^{k-2}$ in the definition of $\tilde{E}'_\ell$ ((5.59)) and polylogarithmic functions $\{p_\ell\}_{\ell=1}^{k-1}$ in the definition of $\tilde{E}_\ell$ ((5.58)) large enough such that*

$$\widetilde{\mathcal{W}} := \sum_{e\in\tilde{E}} \mathsf{w}_e = O(n^{-\kappa}\mathcal{W}(G)).$$

Proof. Fix $\ell \in \{2,\dots,k-1\}$. Let $v_R^{\ell-1}$ be uniform random on $V_{\ell-1}$ (so R is uniform on its support, see (5.44) and (5.45)). Then by Lemma 5.5.6, for any $j_\ell \in [m_\ell]$, if $p_\ell \in \mathsf{polylog}(n)$ is sufficiently large,

$$\Pr[|\mathcal{K}(\overline{R}_1,\dots,\overline{R}_{\ell-1},A_\ell^{(j_\ell)})| \geq p_\ell] \leq n^{-\kappa}. \tag{5.78}$$

Referring to (5.58), this implies that

$$|\tilde{E}_\ell| \leq n^{-\kappa}|\{(u,v) : u \in V_{\ell-1}^+, v \in L^+(u)\}| = n^{-\kappa}|V_{\ell-1}^+|.$$

For $\ell \in \{2,\dots,k-2\}$, by Corollary 5.5.7, if c_ℓ is a sufficiently large constant,

$$\Pr\left[|\mathcal{K}(\overline{R}_1,\dots,\overline{R}_{\ell-1},A_\ell^{(j_\ell)},\overline{R}_{\ell+1})| \geq c_\ell\right] \leq n^{-\kappa},$$

which implies that $|\mathcal{I}(v_{R,j_\ell})| \geq c_\ell$ with probability at most $n^{-\kappa}$ (see (5.52) and (5.53)). Referring to (5.59), this implies that

$$|\tilde{E}'_\ell| \leq n^{-\kappa}|\{(u,v) : u \in V_{\ell-1}^+, v \in L^+(u)\}| = n^{-\kappa}|V_{\ell-1}^+|.$$

Since $\tilde{E}_{k-1} = \emptyset$, the above also holds for $\ell = k-1$.

Combining these, and using the definition of $\tilde{E}$ in (5.63), and the fact that $|V_{\ell-1}^+| = \Theta(|E_\ell| + |\tilde{E}'_\ell|)$, since each vertex in $V_{\ell-1}^+$ has constant out-degree, we have

$$\widetilde{\mathcal{W}} = \sum_{\ell=2}^{k-1} \mathsf{w}_\ell |V_{\ell-1}^+| \leq 2n^{-\kappa}\sum_{\ell=2}^{k-1}\mathsf{w}_\ell O(|E_\ell| + |\tilde{E}'_\ell|) = O\left(n^{-\kappa}\sum_{e\in\overrightarrow{E}(G)}\mathsf{w}_e\right) = O(n^{-\kappa}\mathcal{W}(G)). \square$$

Lemma 5.5.13. *There is a uniform subroutine $\mathcal{S}_k$ such that for all $(u,v) \in E_k$ with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$, $\mathcal{S}_k$ maps $|u,i\rangle$ to $|v,j\rangle$ with error 0 in complexity $\mathsf{T}_{u,v} = \mathsf{T}_k = \widetilde{O}(1)$.*

Proof. The proof is identical to that of Lemma 5.4.5. □

We combine the results of this section into the following:

Corollary 5.5.14. *Let κ be any constant. There is a quantum subroutine (Definition 3.5.1) that implements the full transition map with errors $\epsilon_e \leq n^{-\kappa}$ for all $e \in \overrightarrow{E}(G) \setminus \tilde{E}$, and times: $\mathsf{T}_e = \mathsf{T}_1 = \widetilde{O}(1)$ for all $e \in E_1$; $\mathsf{T}_e = \mathsf{T}_\ell^+ = \widetilde{O}(1)$ for all $e \in E_\ell^+$, for all $\ell \in [k-2]_0$; $\mathsf{T}_e = \mathsf{T}_\ell = \widetilde{O}(\sqrt{n/m_\ell})$ for all $e \in E_\ell$, for all $\ell \in \{2,\dots,k-1\}$; and $\mathsf{T}_e = \mathsf{T}_k = \widetilde{O}(1)$ for all $e \in E_k$.*

Proof. This follows from combining Lemma 5.5.9, Lemma 5.5.10, Lemma 5.5.11 and Lemma 5.5.13 using Lemma 3.5.3. □

5.5.6 Initial state and setup cost

The initial state will be constructed from the uniform superposition over V_0:

$$|\sigma\rangle := \sum_{v_R^0 \in V_0} \frac{1}{\sqrt{|V_0|}} |v_R^0\rangle.$$

Lemma 5.5.15. *The state $|\sigma\rangle$ can be generated with error $O(n^{-\kappa})$ for any constant κ in complexity*

$$\mathsf{S} = \widetilde{O}\left(t_1 + t_2\sqrt{\frac{n}{t_1}} + \cdots + t_{k-1}\sqrt{\frac{n}{t_{k-2}}}\right).$$

Proof. Fix $p \in \mathsf{polylog}(n)$ and a constant c. We start by taking a uniform superposition over all $R_1 \in \binom{A_1}{t_1(2^{c_1}-1)}$ and querying each R_1 to get $D_1(R)$, which costs $\widetilde{O}(t_1)$ (with log factors coming from the cost of inserting everything into data structures as in Section 3.5.1). For $\ell \in \{2, \ldots, k-1\}$, we take a uniform superposition over all sets $R_\ell \in \binom{[m_\ell]}{t_\ell(c_1 \cdots c_{\ell-1}(2^{c_\ell}-1))}$. The total cost so far is $\widetilde{O}(t_1 + t_2 \cdots + t_{k-1})$. Next, we need to populate the rest of the data structure:

For each $\ell \in \{2, \ldots, k-1\}$, do the following.

For each $(s_1, \ldots, s_{\ell-1}, S_\ell) \in [c_1] \times \cdots \times [c_{\ell-1}] \times 2^{[c_\ell]} \setminus \{\emptyset\}$, do the following.

Repeat until pt_ℓ values i_ℓ have been found, or $c \log n$ repetitions have passed in which no i_ℓ was found:

Search for a new value $i_\ell \in R_\ell(s_1, \ldots, s_{\ell-1}, S_\ell)$ such that there exists

$$(i_1, \ldots, i_{\ell-1}, x_{i_\ell}) \in D_{\ell-1}(R_{\ell-1}(s_1, \ldots, s_{\ell-2}, S_{\ell-1}))$$

for some $S_{\ell-1}$ containing $s_{\ell-1}$. If such an i_ℓ is found, insert $(i_1, \ldots, i_\ell, x_{i_\ell})$ into $D_\ell(R_\ell(s_1, \ldots, s_{\ell-1}, S_\ell)$, and increment the forward collision degree of $j_{\ell-1}$ such that $i_{\ell-1} \in A_{\ell-1}^{(j_{\ell-1})}$ stored in $C_{\ell-1}^{\rightarrow}(R)$, as described in the proof of Lemma 5.5.11.

If the inner loop finds $Y \in [pt_\ell]$ values, so $Y = \widetilde{O}(t_\ell)$, it costs at most (up to polylogarithmic factors):

$$\sum_{y=0}^{Y-1} \sqrt{\frac{|R_\ell|}{Y-y}} = \sqrt{\frac{t_\ell n}{m_\ell}} \sum_{y=1}^{Y} \frac{1}{\sqrt{y}} = \Theta\left(\sqrt{\frac{t_\ell n Y}{m_\ell}}\right) = \widetilde{O}\left(t_\ell \sqrt{\frac{n}{m_\ell}}\right),$$

since $|R_\ell| = \Theta(t_\ell n/m_\ell)$. Since $k, c_1, \ldots, c_{k-1}$ are all constant, there are $\widetilde{O}(1)$ loops in total, so the total cost of this procedure is $O(\sqrt{n/m_\ell})$ for a total cost of:

$$\widetilde{O}\left(\sum_{\ell=1}^{k-1} t_\ell + \sum_{\ell=2}^{k-1} t_\ell \sqrt{\frac{n}{m_\ell}}\right) = \widetilde{O}\left(t_1 + \sum_{\ell=1}^{k-2} t_{\ell+1}\sqrt{\frac{n}{t_\ell}}\right),$$

since $t_\ell = \Theta(m_{\ell+1})$ and for all $\ell > 1$, $t_\ell = o(t_1)$ (see Table 5.3).

In parts of the superposition in which there are more than pt_ℓ collisions to be found in some inner loop, we have failed to correctly populate the data $D(R)$, and so the state is not correct. We now argue that this represents a very small part of the state. For uniform random sets $\overline{R}_1, \ldots, \overline{R}_{k-1}$, we could argue that the expected number of ℓ-collisions in $\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_\ell)$ is $\Theta(t_\ell)$, and use a hypergeometric tail inequality to upper bound the proportion of R for which this failure occurs. Things are more complicated, since the sets $\overline{R}_\ell$ for $\ell > 1$ are not uniform on all possible sets – they are composed instead of blocks.

However, by Corollary 5.5.8, for every $\ell \in \{2, \dots, k-1\}$, if $v_{\overline{R}}^{\ell}$ is uniform random on V_ℓ, meaning $R_1, \dots, R_{k-1}$ are uniform random sets, but $\overline{R}_1, \dots, \overline{R}_{k-1}$ have limited support, we still have the necessary tail bound, when c' is a sufficiently large constant:

$$\Pr\left[|\mathcal{K}(\overline{R}_1, \dots, \overline{R}_\ell)| \geq t_\ell c' \log^{2^{\ell-2}}(n)\right] \leq n^{-\kappa}.$$

Thus, choosing $p = c' \log^{2^{\ell-1}}(n)$, the state we generate is $O(n^{-\kappa})$-close to $|\sigma\rangle$. □

5.5.7 Positive analysis

For the positive analysis, we must exhibit a flow (see Definition 3.2.3) on G whenever $M \neq \emptyset$.

Lemma 5.5.16. *There exists some $\mathcal{R}^{\mathsf{T}} = O(|V_0|^{-1})$ such that the following holds. Whenever there is a unique k-collision $(a_1, \dots, a_k) \in A_1 \times \cdots \times A_k$, there exists a flow θ on G that satisfies conditions **P1**-**P5** of Theorem 4.3.1. Specifically:*

1. *For all $e \in \tilde{E}$, $\theta_e = 0$.*
2. *For all $u \in V(G) \setminus (V_0 \cup M)$ and $|\psi_\star\rangle \in \Psi_\star(u)$, $\langle \psi_\star | \theta \rangle = 0$.*
3. *$\sum_{u \in V_0} \theta_u = 1$.*
4. *$\sum_{u \in V_0} \frac{|\theta_u - \sigma(u)|^2}{\sigma(u)} \leq 1$.*
5. *$\mathcal{E}^{\mathsf{T}}(\theta) \leq \mathcal{R}^{\mathsf{T}}$.*

Proof. Recall the definition of M from (5.64). For $\ell \in \{2, \dots, k-1\}$, let $j_\ell^* \in [m_\ell]$ be the unique block label such that $a_\ell \in A_\ell^{(j_\ell^*)}$. Then $a_\ell \in \overline{R}_\ell(s_1, \dots, s_{\ell-1}, S_\ell)$ if and only if $j_\ell^* \in R_\ell(s_1, \dots, s_{\ell-1}, S_\ell)$.

Assuming $M \neq \emptyset$, we define a flow θ on G with all its sinks in M. It will have sources in both V_0 and M, but all other vertices will conserve flow. This will imply **Item 2** for all *correct* star states of G, but we take extra case to ensure that **Item 2** is satisfied for the additional star states in $\Psi_\star(u) : u \in \bigcup_{\ell=0}^{k-2} V_\ell^+$. We define θ from V_0 to V_k as follows.

$\mathcal{R}_0^+$, Item 3, and Item 4: We define M_0 as the set of $v_R^0 \in V_0$ such that for all $\ell \in \{2, \dots, k-1\}$, we have $|\mathcal{K}(\overline{R}_1, \dots, \overline{R}_{\ell-1}, A_\ell^{(j_\ell^*)})| < p_\ell$, where p_ℓ is as in Corollary 5.5.12, and for all $\ell \in \{2, \dots, k-2\}$, $|\mathcal{K}(\overline{R}_1, \dots, \overline{R}_{\ell-1}, A_\ell^{(j_\ell^*)}, \overline{R}_{\ell+1})| = 0$. We define the flow θ over the edges in E_0^+ as

$$\theta_{v_R^0, v_{R,i_1}^0} = \begin{cases} \frac{1}{|M_0|} & \text{if } v_R^0 \in M_0 \text{ and } i_1 = a_1 \\ 0 & \text{else.} \end{cases}$$

That is, each vertex in M_0 has a unique outgoing edge with flow, and the flow is uniformly distributed. From this construction we immediately satisfy **Item 3**.

By Corollary 5.5.7, we know that the proportion of vertices $v_R^0 \in V_0$ that are excluded from M_0 because $|\mathcal{K}(\overline{R}_1, \dots, \overline{R}_{\ell-1}, A_\ell^{(j_\ell^*)}, \overline{R}_{\ell+1})| \geq 1$ is $o(1)$. By (5.78), the proportion of vertices excluded because $|\mathcal{K}(\overline{R}_1, \dots, \overline{R}_{\ell-1}, A_\ell^{(j_\ell^*)})| \geq p_\ell$ is also $o(1)$. Hence, we can compute:

$$\frac{|V_0|}{|M_0|} = \left(1 + O\left(\frac{t_1}{n}\right)\right) \Pi_{\ell=2}^{k-1} \left(1 + O\left(\frac{t_\ell}{m_\ell}\right)\right) (1 + o(1)) = 1 + o(1).$$

Since $\sigma(u) = \frac{1}{|V_0|}$, we can conclude with **Item 4** of the theorem statement:

$$\sum_{u\in V_0} \frac{|\theta_u - \sigma(u)|^2}{\sigma(u)} = |V_0|^2 \left(\frac{1}{|M_0|} - \frac{1}{|V_0|}\right)^2 = \left(\frac{|V_0|}{|M_0|} - 1\right)^2 = o(1).$$

Recall we want to compute $\mathcal{E}^{\mathsf{T}}(\theta) = \mathcal{E}(\theta^{\mathsf{T}})$ (see Definition 3.2.7), which treats an edge e as a path of length T_e. Using $\mathsf{T}_0^+ = \widetilde{O}(1)$ and $\mathsf{w}_0^+ = 1$ (refer to Table 5.5), the contribution of the edges in E_0^+ to the energy of the flow can be computed as

$$\mathcal{R}_0^+ = \sum_{e\in E_0^+} \mathsf{T}_0^+ \frac{\theta_e^2}{\mathsf{w}_0^+} = \widetilde{O}\left(\sum_{u\in M_0} \frac{1}{|M_0|^2}\right) = \widetilde{O}\left(\frac{1}{|M_0|}\right), \tag{5.79}$$

since each vertex in M_0 has a unique outgoing edge with flow and the flow is uniformly distributed.

$\mathcal{R}_1$ and Item 2 (partially): Let M_0^+ be the set of $v_{R,i_1}^0 \in V_0^+$ such that $v_R^0 \in M_0$ and $i_1 = a_1$, so $\left|M_0^+\right| = |M_0|$. These are the only vertices in V_0^+ that have incoming flow, which is equal to $\frac{1}{|M_0|}$. Note that no fault can occur when we add a_1 to R_1 because we have ensured that $a_2 \notin \overline{R}_2$; that is $\mathcal{I}(v_{R,a_1}^0) = \emptyset$, and so by Lemma 5.5.3, $\mathsf{w}_{v_{R,a_1}^0,S_1} = \mathsf{w}_1 = 1$ for all $S_1 \in 2^{[c_1]} \setminus \{\emptyset\}$. To ensure that we satisfy **Item 2** we define the flow as

$$\theta_{v_{R,i_1}^0, v_{R'}^1} = \begin{cases} (-1)^{|S_1|+1}\frac{1}{|M_0|} & \text{if } v_{R,i_1}^0 \in M_0^+ \text{ and } v_{R'}^1 = f_{v_{R,i_1}^0}(S_1), \\ 0 & \text{else} \end{cases}$$

where we recall that $v_{R'}^1 = f_{v_{R,i_1}^0}(S_1)$ if and only if R' is obtained from R by inserting i_1 into $R_1(S_1)$. We verify that indeed for each $u = v_{R,a_1}^0 \in M_0^+$ and $|\psi_\star^{\mathcal{I}_1}(u)\rangle \in \Psi_\star(u)$ (see (5.69)) Item 2 holds:

$$\begin{aligned} \Theta_\star(\mathcal{I}_1, u) &:= \sum_{i\in L^+(u)} \frac{\theta_{u,f_u(i)}}{\sqrt{\mathsf{w}_1}} \langle \psi_\star^{\mathcal{I}_1}(u)|u,i\rangle - \sum_{i\in L^-(u)} \frac{\theta_{u,f_u(i)}}{\sqrt{\mathsf{w}_0^+}} \langle \psi_\star^{\mathcal{I}_1}(u)|u,i\rangle \\ &= \sum_{S_1\in 2^{[c_1]}\setminus\mathcal{I}_1\setminus\{\emptyset\}} \frac{\theta_{u,f_u(S_1)}}{\sqrt{\mathsf{w}_1}}\sqrt{\mathsf{w}_1} + \frac{\theta_{u,f_u(\leftarrow)}}{\sqrt{\mathsf{w}_0^+}}\sqrt{\mathsf{w}_0^+}. \end{aligned} \tag{5.80}$$

We have $f_u(\leftarrow) = v_R^0 \in V_0$, and $\theta_{v_{R,a_1}^0, v_R^0} = -\theta_{v_R^0, v_{R,a_1}^0} = -|M_0|^{-1}$, and $\theta_{u,f_u(S_1)} = (-1)^{|S_1|}|M_0|^{-1}$, so we continue from above:

$$\begin{aligned} \Theta_\star(\mathcal{I}_1, u) &= \sum_{S_1\in 2^{[c_1]}\setminus\mathcal{I}_1\setminus\{\emptyset\}} (-1)^{|S_1|+1}|M_0|^{-1} - (-|M_0|^{-1})(-1) \\ &= -|M_0|^{-1}\left(\sum_{S_1\in 2^{[c_1]}\setminus\mathcal{I}_1} (-1)^{|S_1|} - 1 + 1\right) = 0, \end{aligned} \tag{5.81}$$

since $\sum_{S_1\in 2^{[c_1]}\setminus\mathcal{I}_1}(-1)^{|S_1|} = 0$ (i.e. for any set S, exactly half of its subsets have even size). Using $\mathsf{T}_1 = \widetilde{O}(1)$ and $\mathsf{w}_1 = 1$, the contribution of the edges in E_1 to the energy of the flow can be upper bounded as

$$\mathcal{R}_1 = \sum_{e\in E_1} \mathsf{T}_1 \frac{\theta_e^2}{\mathsf{w}_1} = \widetilde{O}\left(\sum_{u\in M_0^+, S_1\in 2^{[c_1]}\setminus\{\emptyset\}} \frac{1}{|M_0|^2}\right) = \widetilde{O}\left(\frac{1}{|M_0|}\right). \tag{5.82}$$

$\mathcal{R}_\ell^+$ **for** $\ell \in [k-2]$**:** Let $M_\ell(S_1,\dots,S_\ell)$ be the set of $v_R^\ell \in V_\ell(S_1,\dots,S_\ell)$ (see (5.44)) such that $a_1 \in R_1(S_1)$, for all $\ell' \in \{2,\dots,\ell\}$, $j_\ell^* \in R_\ell(\mu(S_1),\dots,\mu(S_{\ell-1}),S_\ell)$, and

$$v^0_{R_1\setminus\{a_1\},R_2\setminus\{j_1^*\},\dots,R_\ell\setminus\{j_\ell^*\},R_{\ell+1},\dots,R_{k-1}} \in M_0.$$

Then letting M_ℓ be the union of all $M_\ell(S_1,\dots,S_\ell)$, we have $|M_\ell| = \Theta(|M_0|)$. We will define θ so that M_ℓ are exactly the vertices of V_ℓ that have non-zero flow coming in from $V_{\ell-1}^+$, and specifically, we will ensure that the amount of incoming flow for each $v_R^\ell \in M_\ell(S_1,\dots,S_\ell)$ is $(-1)^{|S_1|+\cdots+|S_\ell|+\ell}|M_0|^{-1}$. So far this can only be verified for $\ell = 1$ due to the flow that we constructed on E_1, but it will follow for all $\ell \in \{2,\dots,k-1\}$ when we define the flow on E_ℓ (see (5.84)). For now, we define the flow θ over the edges in E_ℓ^+ as

$$\theta_{v_R^\ell, v^\ell_{R,j_\ell}} = \begin{cases} (-1)^{|S_1|+\cdots+|S_\ell|+\ell}\frac{1}{|M_0|} & \text{if } v_R^\ell \in M_\ell(S_1,\dots,S_\ell) \text{ and } j_\ell = j_\ell^*, \\ 0 & \text{else,} \end{cases}$$

so we are just forwarding all flow from v_R^ℓ to a unique neighbour v^ℓ_{R,j_ℓ^*}. Using $\mathsf{T}_\ell^+ = \widetilde{O}(1)$ and $\mathsf{w}_\ell^+ = 1$, the contribution of the edges in E_ℓ^+ to the energy of the flow can be upper bounded as

$$\mathcal{R}_\ell^+ = \sum_{e\in E_\ell^+} \mathsf{T}_\ell^+ \frac{\theta_e^2}{\mathsf{w}_\ell^+} = \widetilde{O}\left(\sum_{u\in M_\ell^+} \frac{1}{|M_0|^2}\right) = \widetilde{O}\left(\frac{1}{|M_0|}\right). \tag{5.83}$$

$\mathcal{R}_\ell$ **for** $\ell \in \{2,\dots,k-1\}$ **and Item 2 (continued):** Let $M_{\ell-1}^+(S_1,\dots,S_{\ell-1})$ be the set of $v^{\ell-1}_{R,j_\ell^*} \in V_{\ell-1}^+(S_1,\dots,S_{\ell-1})$ such that $v_R^{\ell-1} \in M_{\ell-1}(S_1,\dots,S_{\ell-1})$, so letting $M_{\ell-1}^+$ be the union over all the sets $M_{\ell-1}^+(S_1,\dots,S_{\ell-1})$, $\left|M_{\ell-1}^+\right| = O\left(|M_{\ell-1}|\right) = O\left(|M_0|\right)$. Note that $M_{\ell-1}^+(S_1,\dots,S_{\ell-1})$ are exactly the vertices of $V_{\ell-1}^+$ that have non-zero flow coming in from $M_{\ell-1}(S_1,\dots,S_\ell)$. For any $v^{\ell-1}_{R,j_\ell^*} \in M_{\ell-1}^+(S_1,\dots,S_{\ell-1})$, this flow is equal to $(-1)^{|S_1|+\cdots+|S_{\ell-1}|+(\ell-1)}|M_0|^{-1}$. Note that no fault can occur when we add j_ℓ^* to R, because we have ensured in our definition of M_0 that $\mathcal{K}(\overline{R}_1,\dots,\overline{R}_{\ell-1},A_\ell^{(j_\ell^*)},\overline{R}_{\ell+1}) = \emptyset$, so we have $\mathcal{I}(v^{\ell-1}_{R,j_\ell^*}) = \emptyset$, so by (5.60), there is an edge for each $S_\ell \in 2^{[c_\ell]}\setminus\{\emptyset\}$ to which we can assign flow. To ensure that we satisfy **Item 2** we define the flow as

$$\theta_{v^{\ell-1}_{R,j_\ell}, v^\ell_{R'}} = \begin{cases} (-1)^{|S_1|+\cdots+|S_\ell|+\ell}\frac{1}{|M_0|} & \text{if } v_{R,j_\ell} \in M_{\ell-1}^+(S_1,\dots,S_{\ell-1}) \text{ and } v_{R'}^\ell = f_{v^{\ell-1}_{R,j_\ell}}(S_\ell), \\ 0 & \text{else,} \end{cases} \tag{5.84}$$

where we recall that for $v^{\ell-1}_{R,j_\ell} \in V_{\ell-1}^+(S_1,\dots,S_{\ell-1})$, $v_{R'}^\ell = f_{v^{\ell-1}_{R,j_\ell}}(S_\ell)$ if and only if R' is obtained from R by inserting j_ℓ into $R_\ell(\mu(S_1),\dots,\mu(S_{\ell-1}),S_\ell)$. Note that this is consistent with the incoming flow we assumed when defining θ on the edges in $E_{\ell-1}^+$, above. We verify that for each $u = v^{\ell-1}_{R,j_\ell^*} \in M_{\ell-1}^+(S_1,\dots,S_{\ell-1})$ and $|\psi_\star^{\mathcal{I}_\ell}(u)\rangle \in \Psi_\star(u)$ (see (5.70)), Item 2 holds. By a computation nearly identical to (5.80) and (5.81), we obtain:

$$\sum_{i\in L^+(u)} \frac{\theta_{u,f_u(i)}}{\sqrt{\mathsf{w}_\ell}}\langle\psi_\star^{\mathcal{I}_\ell}(u)|u,i\rangle - \sum_{i\in L^-(u)} \frac{\theta_{u,f_u(i)}}{\sqrt{\mathsf{w}_{\ell-1}^+}}\langle\psi_\star^{\mathcal{I}_\ell}(u)|u,i\rangle$$

$$= (-1)^{|S_1|+\cdots+|S_{\ell-1}|+\ell}\left(\sum_{S_\ell\in 2^{[c_\ell]}\setminus\mathcal{I}_\ell} (-1)^{|S_\ell|} - 1 - (-1)\right) = 0.$$

Using $\mathsf{T}_\ell = \widetilde{O}(\sqrt{n/m_\ell})$ and $\mathsf{w}_\ell = \sqrt{n/m_\ell}$ (see Table 5.5), we can upper bound the contribution of the edges in E_ℓ to the energy of the flow:

$$\mathcal{R}_\ell = \sum_{e\in E_\ell} \mathsf{T}_\ell \frac{\theta_e^2}{\mathsf{w}_\ell} = \widetilde{O}\left(\sum_{u\in M^+_{\ell-1}, S_\ell \in 2^{[c_\ell]}\setminus\{\emptyset\}} \frac{1}{|M_0|}\right) = \widetilde{O}\left(\frac{1}{|M_0|}\right). \tag{5.85}$$

$\mathcal{R}_k$: Finally, let $M_{k-1}(S_1,\dots,S_{k-1})$ be the set of $v_R^{k-1} \in V_{k-1}(S_1,\dots,S_{k-1})$ (see (5.44)) such that $a_1 \in R_1(S_1)$, for all $\ell \in \{2,\dots,k-1\}$, $j^*_\ell \in R_\ell(\mu(S_1),\dots,\mu(S_{\ell-1}),S_\ell)$, and $v^0_{R_1\setminus\{a_1\},R_2\setminus\{j^*_2\},\dots,R_{k-1}\setminus\{j^*_{k-1}\}} \in M_0$. We let M_{k-1} be the union of all $M_{k-1}(S_1,\dots,S_{k-1})$. These are exactly the vertices of V_{k-1} that have non-zero incoming flow, with the amount of incoming flow equal to $(-1)^{|S_1|+\cdots+|S_{k-1}|+(k-1)}|M_0|^{-1}$. We define the flow θ on the edges in E_k as

$$\theta_{v_R^{k-1},v^k_{R,i_k}} = \begin{cases} (-1)^{|S_1|+\cdots+|S_{k-1}|+(k-1)}\frac{1}{|M_0|} & \text{if } v_R^k \in M_{k-1}(S_1,\dots,S_{k-1}) \text{ and } i_k = a_k \\ 0 & \text{else.}\end{cases}$$

It is easy to verify that the only vertices $v^k_{R,i_k} \in V_k$ that have non-zero flow are those in M, and thus, all sources and sinks are in $V_0 \cup M$ (M contains some sources, because some vertices have negative flow coming in). Using $\mathsf{T}_k = \widetilde{O}(1)$ and $\mathsf{w}_k = 1$, the contribution of the edges in E_k to the energy of the flow is:

$$\mathcal{R}_k = \sum_{e\in E_k} \mathsf{T}_k \frac{\theta_e^2}{\mathsf{w}_k} = \widetilde{O}\left(\sum_{u\in M_k} \frac{1}{|M_k|^2}\right) = \widetilde{O}\left(\frac{1}{|M_0|}\right). \tag{5.86}$$

Item 1: Recall that $\tilde{E} := \bigcup_{\ell=2}^{k-1}(\tilde{E}_\ell \cup \tilde{E}'_\ell)$ ((5.63)), where $\tilde{E}_\ell \subset E_\ell$ ((5.58)). By ensuring that there is only flow on $v^\ell_{R_1,\dots,R_{k-1}}$ whenever $\mathcal{K}(\overline{R}_1,\dots,\overline{R}_{\ell-1},A_\ell^{(j^*_\ell)})$ is not too big, we have ensured that the flow on the edges in $\bigcup_{\ell=2}^{k-1}\tilde{E}_\ell$ is 0, and by only sending flow down edges that are part of E_ℓ, which is disjoint from $\tilde{E}_{\ell'}$ ((5.59)), the flow on $\bigcup_{\ell=2}^{k-1}\tilde{E}_{\ell'}$ is 0 as well, which implies that the flow on all of $\tilde{E}$ is 0.

Item 5: It remains only to upper bound the energy of the flow by adding up the contributions in (5.79), (5.82), (5.83), (5.85) and (5.86):

$$\mathcal{E}^{\mathsf{T}}(\theta) = \mathcal{R}_0^+ + \mathcal{R}_1 + \sum_{\ell=1}^{k-2}\mathcal{R}_\ell^+ + \sum_{\ell=2}^{k-1}\mathcal{R}_\ell + \mathcal{R}_k = \widetilde{O}\left(\frac{1}{|M_0|}\right).$$

Substituting $|M_0| = \Theta(|V_0|)$ yields the desired upper bound. □

5.5.8 Negative analysis

For the negative analysis, we need to upper bound the total weight of the graph, taking into account the subroutine complexities: $\mathcal{W}^{\mathsf{T}}(G)$.

Lemma 5.5.17. *There exists $\mathcal{W}^{\mathsf{T}}$ such that*

$$\mathcal{W}^{\mathsf{T}}(G) \le \mathcal{W}^{\mathsf{T}} \le \widetilde{O}\left(\left(n + \sum_{\ell=1}^{k-1}\frac{n^2}{t_\ell}\right)|V_0|\right).$$

Proof. Recall that $\mathcal{W}^{\mathsf{T}}(G) = \mathcal{W}(G^{\mathsf{T}})$ is the total weight of the graph G^{T}, where we replace each edge e of G, with weight w_e, by a path of T_e edges of weight w_e, where T_e is the complexity of the edge transition e (see Definition 3.2.7 and **TS1-2** of Theorem 4.3.1). Thus, $\mathcal{W}^{\mathsf{T}}(G) = \sum_{e\in E(G)} \mathsf{T}_e\mathsf{w}_e$. By Corollary 5.5.14 (see also Table 5.5) $\mathsf{T}_e = \widetilde{O}(1)$ for all $e \in E_1 \cup E_k \cup \bigcup_{\ell\in[k-2]_0} E_\ell^+$ and $\mathsf{T}_e = \widetilde{O}(\sqrt{n/m_\ell})$ for all $e \in E_\ell$ for $\ell \in \{2,\dots,k-1\}$. We have defined the weight function (see Table 5.5) so that $\mathsf{w}_e = 1$ for all $e \in E_1 \cup E_k \cup \bigcup_{\ell\in[k-2]_0} E_\ell^+$ and $\mathsf{w}_e = \sqrt{n/m_\ell}$ for all $e \in E_\ell$ for $\ell \in \{2,\dots,k-1\}$. Thus, using (5.49), the total contribution to the weight from the edges in E_0^+ is:

$$\mathcal{W}_0^+ := \mathsf{T}_0^+\mathsf{w}_0^+ \left|E_0^+\right| = \widetilde{O}\left(n\left|V_0\right|\right). \tag{5.87}$$

For $\ell \in [k-2]$, we can use (5.50) to compute the total contribution to the weight from the edges in E_ℓ^+:

$$\mathcal{W}_\ell^+ := \mathsf{T}_\ell^+\mathsf{w}_\ell^+ \left|E_\ell^+\right| = \widetilde{O}\left(n\left|V_0\right|\right). \tag{5.88}$$

Using (5.56), the total contribution from the edges in E_1 is:

$$\mathcal{W}_1 := \mathsf{T}_1\mathsf{w}_1 \left|E_1\right| = \widetilde{O}\left(n\left|V_0\right|\right). \tag{5.89}$$

For $\ell \in \{2,\dots,k-1\}$ using (5.61) the total contribution from the edges in E_ℓ is:

$$\mathcal{W}_\ell := \mathsf{T}_\ell\mathsf{w}_\ell \left|E_\ell\right| = \widetilde{O}\left(\frac{n}{m_\ell}\right)\left|E_\ell\right| = \widetilde{O}\left(\frac{n^2}{m_\ell}\left|V_0\right|\right). \tag{5.90}$$

Finally, using (5.62), the total contribution from the edges in E_k is:

$$\mathcal{W}_k := \mathsf{T}_k\mathsf{w}_k \left|E_k\right| = \widetilde{O}\left(\frac{n^2}{t_{k-1}}\left|V_0\right|\right). \tag{5.91}$$

Combining (5.87) to (5.91), we get total weight:

$$\mathcal{W}^{\mathsf{T}}(G) = \widetilde{O}\left(\left(n + \sum_{\ell=1}^{k-2} n + n + \sum_{\ell=2}^{k-1}\frac{n^2}{m_\ell} + \frac{n^2}{t_{k-1}}\right)\left|V_0\right|\right) = \widetilde{O}\left(\left(n + \sum_{\ell=1}^{k-1}\frac{n^2}{t_\ell}\right)\left|V_0\right|\right),$$

using $m_\ell = \Theta(t_{\ell-1})$ for all $\ell \in \{2,\dots,k-1\}$. □

5.5.9 Conclusion of proof of Theorem 5.5.1

We can now conclude with the proof of Theorem 5.5.1, showing an upper bound of $\widetilde{O}\left(n^{\frac{3}{4}-\frac{1}{4}\frac{1}{2^{k-1}}}\right)$ on the bounded error quantum time complexity of k-distinctness.

Proof of Theorem 5.5.1. We apply Theorem 4.3.1 to G (Section 5.5.2 and Section 5.5.1), M ((5.64)), σ the uniform distribution on V_0 ((5.42)), and $\Psi_\star$ (Section 5.5.3), with

$$\mathcal{W}^{\mathsf{T}} = \widetilde{O}\left(\left(n + \sum_{\ell=1}^{k-1}\frac{n^2}{t_\ell}\right)\left|V_0\right|\right) \text{ and } \mathcal{R}^{\mathsf{T}} = \widetilde{O}\left(\left|V_0\right|^{-1}\right).$$

Then we have, referring to Table 5.3,

$$\mathcal{W}^{\mathsf{T}}\mathcal{R}^{\mathsf{T}} = \widetilde{O}\left(n + \sum_{\ell=1}^{k-1}\frac{n^2}{t_\ell}\right) = o(n^2).$$

We have shown the following:

Setup Subroutine: By Lemma 5.5.15, the state $|\sigma\rangle$ can be generated in cost

$$\mathsf{S} = \widetilde{O}\left(t_1 + \sum_{\ell=1}^{k-2} t_{\ell+1}\sqrt{\frac{n}{t_\ell}}\right).$$

Star State Generation Subroutine: By Lemma 5.5.5, the star states $\Psi_\star$ can be generated in $\widetilde{O}(1)$ complexity.

Transition Subroutine: By Corollary 5.5.14, there is a quantum subroutine that implements the transition map with errors $\epsilon_{u,v}$ and costs $\mathsf{T}_{u,v}$, such that

TS1 For all $(u,v) \in \overrightarrow{E}(G) \setminus \tilde{E}$ (defined in (5.63)), taking $\kappa > 2$ in Lemma 5.5.11, we have $\epsilon_{u,v} = O(n^{-\kappa}) = o(1/(\mathcal{R}^{\mathsf{T}}\mathcal{W}^{\mathsf{T}}))$.

TS2 By Corollary 5.5.12, Lemma 5.5.17 and using $\kappa > 2$:

$$\widetilde{\mathcal{W}} = O(n^{-\kappa}\mathcal{W}^{\mathsf{T}}(G)) = o(1/\mathcal{R}^{\mathsf{T}}).$$

Checking Subroutine: By (5.65), for any $u \in V_{\mathsf{M}} = V_k$, we can check if $u \in M$ in cost $\widetilde{O}(1)$.

Positive Condition: By Lemma 5.5.16, there exists a flow satisfying conditions **P1**-**P5** of Theorem 4.3.1, with $\mathcal{E}^{\mathsf{T}}(\theta) \leq \mathcal{R}^{\mathsf{T}} = \widetilde{O}\left(|V_0|^{-1}\right)$.

Negative Condition: By Lemma 5.5.17, $\mathcal{W}^{\mathsf{T}}(G) \leq \mathcal{W}^{\mathsf{T}} = \widetilde{O}\left(\left(n + \sum_{\ell=1}^{k-1} \frac{n^2}{t_\ell}\right)|V_0|\right)$.

Thus, by Theorem 4.3.1, there is a quantum algorithm that decides if $M = \emptyset$ in bounded error in complexity:

$$\begin{aligned}\widetilde{O}\left(\mathsf{S} + \sqrt{\mathcal{R}^{\mathsf{T}}\mathcal{W}^{\mathsf{T}}}\right) &= \widetilde{O}\left(t_1 + \sum_{\ell=1}^{k-2} t_{\ell+1}\sqrt{\frac{n}{t_\ell}} + \sqrt{n} + \sum_{\ell=1}^{k-1}\frac{n}{\sqrt{t_\ell}}\right)\\ &= \widetilde{O}\left(t_1 + \sum_{\ell=1}^{k-2} t_{\ell+1}\sqrt{\frac{n}{t_\ell}} + \sqrt{n} + \frac{n}{\sqrt{t_{k-1}}}\right)\end{aligned}$$

since $t_1 > t_2 > \cdots > t_{k-1}$. Choosing the optimal values of $t_\ell = n^{\frac{3}{4} - \frac{1}{4}\frac{1}{2^{k}-1} - \sum_{\ell'=2}^{\ell}\frac{2^{k-1-\ell'}}{2^k-1}}$ for $\ell \in [k-1]$, as in Table 5.3, we get an upper bound of $\widetilde{O}\left(n^{\frac{3}{4}-\frac{1}{4}\frac{1}{2^{k}-1}}\right)$. Since $M \neq \emptyset$ if x has a unique k-collision, and $M = \emptyset$ if x has no k-collision, the algorithm distinguishes these two cases. By Lemma 5.3.1, this is enough to decide k-distinctness in general. □

Chapter 6

Multidimensional electrical networks

World goin' one way, people another.

Poot, The Wire

This chapter is based on the paper *Multidimensional Electrical Networks and their Application to Exponential Speedups for Graph Problems* [LZ23], which is joint work with Jianqiang Li.

While the multidimensional quantum walk framework is capable of achieving exponential speedups, its connection to electrical networks remains unclear due to its generalisation compared to existing quantum walks. In this chapter, we reestablish this connection, which we call the multidimensional electrical network framework. By generalising existing electrical networks through the use of alternative neighbourhoods, this framework enables the generation of quantum states corresponding to an alternative electrical flow over the edges of graphs, which admits an alternative potential. We argue that these definitions are natural by demonstrating that the alternative flow and potential satisfy the same linear relationships as the standard flow and potential in electrical networks. We first apply this framework to locate a marked vertex in one-dimensional random hierarchical graphs. Additionally, we define a family of graphs where we can efficiently generate the quantum alternative electrical flow state and sample from it to recover an s-t path exponentially faster than is possible classically.

6.1 Sampling from the electrical flow

In Section 3.2, we discussed quantum walks, electrical networks, and the connection between them. More specifically, we saw how this connection led to a quantum walk algorithm that approximates a quantum state representing the electrical flow in a network, as described in Corollary 3.4.2. In Chapter 4, we explored how to generalise quantum walks through the multidimensional quantum walk framework. To preserve the aforementioned connection after this generalisation, we also need to extend the notion of an electrical network. In this chapter, we construct this generalisation, which we call the *multidimensional electrical network*. We demonstrate how it naturally arises by generalising Kirchhoff's Law and Ohm's Law into Alternative Kirchhoff's Law and Alternative Ohm's Law, respectively. This generalisation is achieved by incorporating the alternative neighbourhoods technique from Section 4.2 into these electrical network laws. Before tackling this generalisation, however, we first discuss the importance of efficiently approximating the electrical flow state.

In Corollary 3.4.1, we discussed how to design a quantum walk algorithm that detects the existence of marked vertices in a graph. In [AGJ20], it was shown how to extend this result to *find* a marked vertex. An alternative approach was proposed in [Pid19], which works by sampling from (an approximation of) the electrical flow state. By adding new edges from the marked vertices to a new vertex t_0, an s-t_0 electrical flow θ is created in the modified network. Assigning appropriate weights to these new edges ensures that, by sampling from the electrical flow state $|\theta\rangle$, one can obtain one of these new edges (each containing a marked vertex) with high probability. A key subroutine of this algorithm, which is interesting in its own right, approximates the effective resistance $\mathcal{R}_{s,t}$ in the electrical network G. By extending the connection between quantum walks and electrical networks to the multidimensional quantum walk framework, we open up the possibility of integrating the alternative neighbourhoods technique with these applications.

As seen in (3.6), another perspective on electrical flow is that it forms the solution to a Laplacian linear system. Classically, such solutions can be approximately computed in nearly-linear time (in the number of edges) [ST14]. However, as in the case of welded trees graphs discussed in Section 4.4, this becomes impractical if the network contains an exponential number of edges. By combining the Laplacian linear system with the concept of alternative neighbourhoods, we show in this chapter how, in certain cases, we can approximate the electrical flow in an exponentially large network in polynomial time. It is important to note, however, that this solution approximates the exponentially large quantum state $|\theta\rangle$. As with the HHL algorithm [HHL09], fully extracting the solution into its classical form would still require exponential time.

6.2 Multidimensional electrical networks

6.2.1 Alternative neighbourhoods revisited

Recall from Section 4.2 that multidimensional quantum walks modify the quantum walk operator through the use of alternative neighbourhoods (see Definition 4.2.1), resulting in the operator

$$U_{\mathcal{A}^{\mathsf{alt}}\mathcal{B}} = (2\Pi_{\mathcal{A}^{\mathsf{alt}}} - I)(2\Pi_{\mathcal{B}} - I),$$

where $\Pi_{\mathcal{A}^{\mathsf{alt}}}$ is the orthogonal projector onto $\mathcal{A}^{\mathsf{alt}}$, meaning

$$2\Pi_{\mathcal{A}^{\mathsf{alt}}} - I = 2 \sum_{u \in V(G')\setminus\{s_0,t\}} \sum_{i=0}^{a_u - 1} |\overline{\psi}_{u,i}\rangle\langle\overline{\psi}_{u,i}| - I.$$

To distinguish this from the case without alternative neighbourhoods and from "regular" electrical networks, we retain the superscript alt throughout this chapter.

As in Section 4.2, we "walk" on the modified network G', which is obtained by connecting the vertex s_0 to s via an edge with weight w_0, as shown in Figure 3.2. Hence our phase estimation algorithm operates on the Hilbert space

$$\mathcal{H} = \operatorname{span}\{|u, v\rangle : (u, v) \in E(G')\}. \tag{6.1}$$

The initial motivation for our multidimensional electrical network, is to be able to apply Lemma 3.3.11 to this more general walk operator as well, resulting in a variant of Corollary 3.4.2 that allows for the use of alternative neighbourhoods. More concretely, this means that we need to find a new optimal positive witness $|w^{\mathsf{alt}}\rangle$ (see Definition 3.3.9) with respect to $\Psi^{\mathcal{A}^{\mathsf{alt}}}$ instead of Ψ^{A} and an (unnormalised) state $|w_{\mathcal{A}^{\mathsf{alt}}}\rangle \in \mathcal{A}^{\mathsf{alt}}$ such that

$$|\psi_0\rangle = \sqrt{p}\frac{|w^{\mathsf{alt}}\rangle}{\||w^{\mathsf{alt}}\rangle\|} + (I - \Pi_{\mathcal{B}})|w^{\mathsf{alt}}_{\mathcal{A}}\rangle.$$

For simplicity we will assume in the rest of this work that s and t do not contain any additional alternative neighbourhoods, as it greatly simplifies notation and intuition. In our applications in Section 6.3 and Section 6.4 these simplifying assumptions will also hold.

6.2.2 Alternative Kirchhoff's Law

We begin with the construction of $|w^{\mathsf{alt}}\rangle$, which, as in the case without alternative neighbourhoods, starts with a flow state (see (3.29)). For any unit flow θ, we have that $|\theta\rangle$ lies in the symmetric subspace $\mathcal{B}^{\perp}$ by construction. However, our optimal witness $|w^{\mathsf{alt}}\rangle$, with respect to $\Psi^{\mathcal{A}^{\mathsf{alt}}}$, must also satisfy $\Pi_{\mathcal{A}^{\mathsf{alt}}}|w^{\mathsf{alt}}\rangle = 0$. In (3.31), we used Kirchhoff's Law to achieve this, showing that for any s-t flow θ and any vertex $u \in V\backslash\{s, t\}$, we have $\langle\psi_\star(u)|\theta\rangle = 0$. However, with the addition of alternative neighbourhoods, being orthogonal to all star states is insufficient to guarantee orthogonality to $\mathcal{A}^{\mathsf{alt}}$, as seen in Section 4.2.1. Instead, the flow state $|\theta\rangle$ must be orthogonal to all of $\operatorname{span}(\Psi_\star(u))$ for every $u \in V\backslash\{s, t\}$. Therefore, we modify Kirchhoff's Law to obtain the *Alternative Kirchhoff's Law*:

Definition 6.2.1 (Alternative Kirchhoff's Law)**.** *For any s-t alternative flow θ^{alt} with respect to a collection of alternative neighbourhoods $\Psi_\star$ on an electrical network $G = (V, E, \mathsf{w})$ with $s, t \in V$, the corresponding flow state $|\theta^{\mathsf{alt}}\rangle$ is orthogonal to $\operatorname{span}(\Psi_\star(u))$ for every $u \in V\backslash\{s, t\}$, that is, $\langle\psi_{u,i}|\theta\rangle = 0$ for each $i \in [a_u - 1]_0$.*

We refer to any unit s-t flow satisfying Alternative Kirchhoff's Law as an *alternative unit s-t flow*. This law can be interpreted as the conservation of flow across all alternative neighbourhoods, which we previously encountered as **P2** in Theorem 4.3.1. Similarly to Definition 3.2.3, we define the s-t alternative electrical flow with respect to $\Psi_\star$ as the alternative unit s-t flow with minimal energy:

Definition 6.2.2 (Alternative Electrical Flow)**.** *For a collection of alternative neighbourhoods $\Psi_\star$ on an electrical network $G = (V, E, \mathsf{w})$ with $s, t \in V$, the* s-t alternative electrical flow *is the alternative unit s-t flow with minimal energy $\mathcal{E}(\theta^{\mathsf{alt}})$. We call this minimal energy the* alternative effective resistance $\mathcal{R}^{\mathsf{alt}}_{s,t}$.

At first glance, it may seem that multiple alternative unit s-t flows could achieve the minimal energy $\mathcal{R}^{\mathsf{alt}}_{s,t}$. However, we will prove in Theorem 6.2.7 that the s-t alternative electrical flow is indeed unique, assuming an alternative unit s-t flow exists. In some

cases, the "regular" s-t electrical flow may also satisfy Alternative Kirchhoff's Law, making it identical to the s-t alternative electrical flow. We present an example of this in Section 6.3, allowing us to apply Lemma 3.3.11 with similar parameters as in Corollary 3.4.2. In contrast, there may be cases where no s-t flow satisfies Alternative Kirchhoff's Law, and thus the s-t alternative electrical flow does not exist. We have encountered such an example in Section 4.2.1. Most commonly, we find ourselves in between these extremes, where the s-t electrical flow and the s-t alternative electrical flow do not coincide, and Ohm's Law does not hold.

Before we address the problem of not being able to apply Ohm's Law, we complete the construction of our optimal positive witness. To obtain a state orthogonal to $\mathcal{A}^{\mathsf{alt}}$, we seek an s_0-t flow $\theta^{\mathsf{alt}\prime}$ on G'. Since by assumption s contains no additional alternative neighbourhoods, we can can construct $\theta^{\mathsf{alt}\prime}$ from any alternative s-t flow θ^{alt} by sending one unit of flow along the edge $(s_0, s) \in \overrightarrow{E}(G')$. According to (3.29), the corresponding flow state becomes

$$\begin{aligned}|\theta^{\mathsf{alt}\prime}\rangle &:= \frac{1}{\sqrt{2\mathcal{E}(\theta^{\mathsf{alt}\prime})}} \sum_{(u,v)\in\overrightarrow{E}(G')} \frac{\theta^{\mathsf{alt}\prime}_{u,v}}{\sqrt{\mathsf{w}_{u,v}}} \left(|u,v\rangle + |v,u\rangle\right) \\ &= \frac{1}{\sqrt{2\left(\mathcal{E}(\theta^{\mathsf{alt}}) + 1/\mathsf{w}_0\right)}} \left(\sum_{(u,v)\in\overrightarrow{E}(G)} \frac{\theta^{\mathsf{alt}}_{u,v}}{\sqrt{\mathsf{w}_{u,v}}} \left(|u,v\rangle + |v,u\rangle\right) + \frac{1}{\mathsf{w}_0}\left(|s_0,s\rangle + |s,s_0\rangle\right) \right). \end{aligned} \tag{6.2}$$

By Alternative Kirchhoff's Law, $|\theta^{\mathsf{alt}\prime}\rangle$ serves as a valid 0-positive witness for every s_0-t alternative unit flow $\theta^{\mathsf{alt}\prime}$. Assuming that the optimal positive witness takes this form, we have, similarly to (3.33) and by Definition 3.3.9, that the optimal positive witness (with respect to $\mathcal{A}^{\mathsf{alt}}$) is given by

$$|w^{\mathsf{alt}}\rangle = \underset{\sqrt{\mathsf{w}_0\mathcal{E}(\theta^{\mathsf{alt}})+1}|\theta^{\mathsf{alt}\prime}\rangle}{\arg\min} \left\{ \left\| \sqrt{\mathsf{w}_0\mathcal{E}(\theta^{\mathsf{alt}})+1}|\theta^{\mathsf{alt}\prime}\rangle \right\| : \theta^{\mathsf{alt}} \text{ is an alternative } s\text{-}t \text{ unit flow} \right\}.$$

Thus, the optimal positive witness is

$$|w^{\mathsf{alt}}\rangle = \sqrt{\mathsf{w}_0\mathcal{R}^{\mathsf{alt}}_{s,t}+1}|\theta^{\mathsf{alt}\prime}\rangle, \tag{6.3}$$

where θ^{alt} is the s-t alternative electrical flow. Similar to the case without alternative neighbourhoods, we set $\mathsf{w}_0 = \frac{1}{\mathcal{R}^{\mathsf{alt}}_{s,t}}$ to ensure that $\frac{|\langle w^{\mathsf{alt}}|\psi_0\rangle|^2}{\||w^{\mathsf{alt}}\rangle\|^2}$ is at least a constant, as required by Theorem 3.3.8. Just as in (3.35), we have

$$\mathcal{R}^{\mathsf{alt}}_{s_0,t} = \sum_{(u,v)\in\overrightarrow{E}(G')} \frac{\left(\theta^{\mathsf{alt}\prime}_{u,v}\right)^2}{\mathsf{w}_{u,v}} = \frac{\left(\theta^{\mathsf{alt}}_{s_0,s}\right)^2}{\mathsf{w}_{s_0,s}} + \sum_{(u,v)\in\overrightarrow{E}(G)} \frac{\left(\theta^{\mathsf{alt}}_{u,v}\right)^2}{\mathsf{w}_{u,v}} = \frac{1}{\mathsf{w}_0} + \mathcal{R}^{\mathsf{alt}}_{s,t} = 2\mathcal{R}^{\mathsf{alt}}_{s,t}, \tag{6.4}$$

which satisfies

$$\frac{|\langle w^{\mathsf{alt}}|\psi_0\rangle|^2}{\||w^{\mathsf{alt}}\rangle\|^2} = \frac{1}{\mathsf{w}_0\mathcal{R}^{\mathsf{alt}}_{s_0,t}} = \frac{1}{2}. \tag{6.5}$$

6.2.3 Alternative Ohm's Law

We have now constructed our optimal positive witness, but we still need to verify our assumption that $|w^{\mathsf{alt}}\rangle$ is always (up to a multiplicative factor) equal to $|\theta^{\mathsf{alt}\prime}\rangle$ for some s-t

alternative unit flow θ^{alt}, which we prove in (6.8). For now, let us return to Lemma 3.3.11, where we still need to find an unnormalised state $|w_{\mathcal{A}^{\mathsf{alt}}}\rangle \in \mathcal{A}^{\mathsf{alt}}$ such that

$$|\psi_0\rangle = \sqrt{p}\frac{|w^{\mathsf{alt}}\rangle}{\||w^{\mathsf{alt}}\rangle\|} + (I - \Pi_{\mathcal{B}})|w^{\mathsf{alt}}_{\mathcal{A}}\rangle.$$

If the s-t alternative electrical flow θ^{alt} does not overlap with the s-t electrical flow, we cannot find a potential vector p defined on the vertices V that satisfies Ohm's Law. Instead, we look for a potential vector $\mathsf{p}^{\mathsf{alt}}$ on the edges E, meaning it assigns a potential $\mathsf{p}^{\mathsf{alt}}_{u,v}$ to each edge $(u,v) \in E$.

Definition 6.2.3 (Alternative Potential)**.** *An* alternative potential vector *(or alternative potential function) on a network $G = (V, E, \mathsf{w})$ is a real-valued function $\mathsf{p}^{\mathsf{alt}} : E \to \mathbb{R}$ that assigns a potential $\mathsf{p}_{u,v}$ to each ordered pair $(u,v) \in E$.*

Similar to how the potential vector satisfies $\mathsf{p}_s = \mathcal{R}_{s,t}$ and $\mathsf{p}_t = 0$, we require the alternative potential vector $\mathsf{p}^{\mathsf{alt}}$ to satisfy $\mathsf{p}^{\mathsf{alt}}_{s_0,v} = \mathcal{R}^{\mathsf{alt}}_{s,t}$ and $\mathsf{p}^{\mathsf{alt}}_{t,v} = 0$ for every $v \in \Gamma(s)$ and $v \in \Gamma(t)$, respectively. The corresponding alternative potential state is defined as

$$|\mathsf{p}^{\mathsf{alt}}\rangle := \sqrt{\frac{2}{\mathcal{R}^{\mathsf{alt}}_{s,t}}} \sum_{V(G)\setminus\{s\}} \sum_{v\in\Gamma_G(u)} \sqrt{\mathsf{w}_{u,v}}(\mathsf{p}^{\mathsf{alt}}_{u,v}|u,v\rangle - \mathsf{p}^{\mathsf{alt}}_{v,u}|v,u\rangle). \tag{6.6}$$

Definition 6.2.4 (Alternative Ohm's Law)**.** *Let θ^{alt} be the s-t alternative electrical flow with respect to a collection of alternative neighbourhoods $\Psi_\star$ on an electrical network $G = (V, E, \mathsf{w})$ with $s, t \in V$. Then the alternative potential vector $\mathsf{p}^{\mathsf{alt}}$ corresponding to θ^{alt} assigns a potential $\mathsf{p}^{\mathsf{alt}}_{u,v}$ on each edge $(u,v) \in E$ such that $\mathsf{p}^{\mathsf{alt}}_{s_0,v} = \mathcal{R}^{\mathsf{alt}}_{s,t}$ and $\mathsf{p}^{\mathsf{alt}}_{t,v} = 0$ for every $v \in \Gamma(s)$ and $v \in \Gamma(t)$, respectively. Additionally, the associated state $|\mathsf{p}^{\mathsf{alt}}\rangle$ (see (6.6)) satisfies $\Pi_{\mathcal{A}^{\mathsf{alt}}}|\mathsf{p}^{\mathsf{alt}}\rangle = |\mathsf{p}^{\mathsf{alt}}\rangle$ and the potential difference between (u,v) and (v,u) is equal to the amount of electrical flow $\theta^{\mathsf{alt}}_{u,v}$ along (u,v) multiplied with the resistance $1/\mathsf{w}_{u,v}$, that is, $\mathsf{p}^{\mathsf{alt}}_{u,v} - \mathsf{p}^{\mathsf{alt}}_{v,u} = \theta^{\mathsf{alt}}_{u,v}/\mathsf{w}_{u,v}$.*

We have not yet introduced the necessary tools to show that an alternative potential vector $\mathsf{p}^{\mathsf{alt}}$ satisfying Alternative Ohm's Law always exists. Without the additional condition $\Pi_{\mathcal{A}^{\mathsf{alt}}}|\mathsf{p}^{\mathsf{alt}}\rangle = |\mathsf{p}^{\mathsf{alt}}\rangle$, there would be no dependency between the variables $\mathsf{p}^{\mathsf{alt}}_{u,v}$ and $\mathsf{p}^{\mathsf{alt}}_{v,u}$ for different edges, and it would always be possible to find a potential vector satisfying Alternative Ohm's Law. However, we will prove in Theorem 6.2.9 that the condition $\Pi_{\mathcal{A}^{\mathsf{alt}}}|\mathsf{p}^{\mathsf{alt}}\rangle = |\mathsf{p}^{\mathsf{alt}}\rangle$ ensures the uniqueness of the alternative potential. The state $|\mathsf{p}^{\mathsf{alt}}\rangle$ will not be our choice of $|w_{\mathcal{A}^{\mathsf{alt}}}\rangle$, but it will in fact be $|\mathsf{p}^{\mathsf{alt}\prime}\rangle$. Here, $\mathsf{p}^{\mathsf{alt}\prime}$ corresponds to the potential of the s_0-t electrical flow on G', with the alternative potential state given by

$$\begin{aligned} |\mathsf{p}^{\mathsf{alt}\prime}\rangle &:= \sqrt{\frac{2}{\mathcal{R}^{\mathsf{alt}}_{s_0,t}}} \sum_{V(G')\setminus\{s_0\}} \sum_{v\in\Gamma_{G'}(u)} (-1)^{\Delta_{u,v}} \mathsf{p}^{\mathsf{alt}\prime}_{u,v} \sqrt{\mathsf{w}_{u,v}}|u,v\rangle \\ &= \sqrt{\frac{1}{\mathcal{R}^{\mathsf{alt}}_{s,t}}} \sum_{V(G)} \sum_{v\in\Gamma_{G'}(u)} (-1)^{\Delta_{u,v}} \mathsf{p}^{\mathsf{alt}\prime}_{u,v} \sqrt{\mathsf{w}_{u,v}}|u,v\rangle. \end{aligned} \tag{6.7}$$

If the potential vector $\mathsf{p}^{\mathsf{alt}\prime}$ satisfies Alternative Ohm's Law, then, using a similar derivation

as in the proof of Corollary 3.4.2, we see that $|\mathsf{p}^{\mathsf{alt}\prime}\rangle$ is a valid candidate for $|w_{\mathcal{A}^{\mathsf{alt}}}\rangle$:

$$
\begin{aligned}
|\theta^{\mathsf{alt}\prime}\rangle &= \frac{1}{2\sqrt{\mathcal{R}^{\mathsf{alt}}_{s_0,t}}} \sum_{(u,v)\in\vec{E}(G')} \frac{\theta^{\mathsf{alt}\prime}_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}(|u,v\rangle + |v,u\rangle) \\
&= \frac{1}{2\sqrt{\mathcal{R}^{\mathsf{alt}}_{s_0,t}}} \sum_{(u,v)\in\vec{E}(G')} (\mathsf{p}^{\mathsf{alt}\prime}_{u,v} - \mathsf{p}^{\mathsf{alt}\prime}_{v,u})\sqrt{\mathsf{w}_{u,v}}\,(|u,v\rangle + |v,u\rangle) \\
&= \frac{1}{2\sqrt{\mathcal{R}^{\mathsf{alt}}_{s_0,t}}}(I + \mathsf{SWAP}) \sum_{V(G')} \sum_{v\in\Gamma_{G'}(u)} (-1)^{\Delta_{u,v}} \mathsf{p}^{\mathsf{alt}\prime}_{u,v}\sqrt{\mathsf{w}_{u,v}}|u,v\rangle \\
&= (I - \Pi_{\mathcal{B}})\frac{1}{\sqrt{\mathcal{R}^{\mathsf{alt}}_{s,t}}} \sum_{V(G')} \sum_{v\in\Gamma_{G'}(u)} (-1)^{\Delta_{u,v}} \mathsf{p}^{\mathsf{alt}\prime}_{u,v}\sqrt{\mathsf{w}_{u,v}}|u,v\rangle && \text{see (3.25)} \\
&= (I - \Pi_{\mathcal{B}})|\mathsf{p}^{\mathsf{alt}\prime}\rangle + (I - \Pi_{\mathcal{B}})\frac{1}{\sqrt{\mathcal{R}^{\mathsf{alt}}_{s,t}}} \sum_{v\in\Gamma(s_0)} (-1)^{\Delta_{s_0,v}} \mathsf{p}^{\mathsf{alt}\prime}_{s_0,v}\sqrt{\mathsf{w}_{s_0,v}}|s_0,v\rangle && \text{see (6.7)} \\
&= (I - \Pi_{\mathcal{B}})|\mathsf{p}^{\mathsf{alt}\prime}\rangle + (I - \Pi_{\mathcal{B}})\frac{1}{\sqrt{\mathcal{R}^{\mathsf{alt}}_{s,t}}} 2\sqrt{\mathcal{R}^{\mathsf{alt}}_{s,t}}|\psi_\star(s_0)\rangle = (I - \Pi_{\mathcal{B}})|\mathsf{p}'\rangle + \sqrt{2}|\psi_0\rangle. && \text{see (3.28)}
\end{aligned}
$$

This can be rewritten as

$$
|\psi_0\rangle = \frac{1}{\sqrt{2}}|\theta^{\mathsf{alt}\prime}\rangle - (I - \Pi_{\mathcal{B}})\frac{1}{\sqrt{2}}|\mathsf{p}^{\mathsf{alt}\prime}\rangle. \tag{6.8}
$$

By Definition 6.2.4, we know that $|\mathsf{p}^{\mathsf{alt}}\rangle \in \mathcal{A}^{\mathsf{alt}}$, showing that our choice of $|w^{\mathsf{alt}}\rangle$ was indeed the optimal witness, as indicated by Fact 3.3.10. Thus, after applying Lemma 3.3.11 with $|w_{\mathcal{A}^{\mathsf{alt}}}\rangle = \frac{1}{\sqrt{2}}|\mathsf{p}^{\mathsf{alt}\prime}\rangle$, the resulting state after running phase estimation on the quantum walk operator $U_{\mathcal{A}^{\mathsf{alt}}\mathcal{B}}$ with the initial state $|\psi_0\rangle$ approximates the s_0-t electrical flow state $|\theta^{\mathsf{alt}\prime}\rangle$. We will formalise this in Theorem 6.2.9 after proving that both θ^{alt} and $\mathsf{p}^{\mathsf{alt}}$ exist and are unique.

In the following examples and applications, we demonstrate this existence by explicitly constructing the state $|\mathsf{p}^{\mathsf{alt}}\rangle$. To verify that $\Pi_{\mathcal{A}^{\mathsf{alt}}}|\mathsf{p}^{\mathsf{alt}}\rangle = |\mathsf{p}^{\mathsf{alt}}\rangle$, we introduce the states:

$$
|\mathsf{p}^{\mathsf{alt}}_{|u}\rangle = \sqrt{\frac{2}{\mathcal{R}^{\mathsf{alt}}_{s,t}}}(|u\rangle\langle u| \otimes I)|\mathsf{p}^{\mathsf{alt}\prime}\rangle
$$

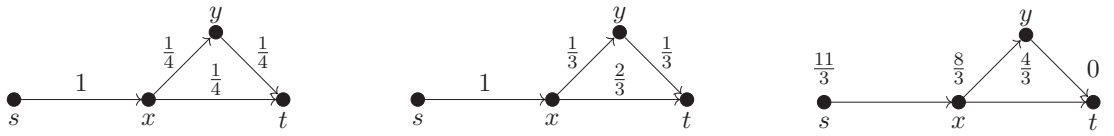
for $u \in V(G)\setminus\{s\}$. To verify whether $\Pi_{\mathcal{A}^{\mathsf{alt}}}|\mathsf{p}^{\mathsf{alt}}\rangle = |\mathsf{p}^{\mathsf{alt}}\rangle$, it suffices to verify that each $|\mathsf{p}^{\mathsf{alt}}_{|u}\rangle$ lies in $\mathrm{span}\{\Psi_\star(u)\}$, since we can decompose $|\mathsf{p}^{\mathsf{alt}}\rangle$ as

$$
\begin{aligned}
|\mathsf{p}^{\mathsf{alt}}\rangle &= \sqrt{\frac{2}{\mathcal{R}^{\mathsf{alt}}_{s,t}}} \sum_{u\in V(G)\setminus\{s\}} \sum_{v\in\Gamma_G(u)} (-1)^{\Delta_{u,v}} \mathsf{p}^{\mathsf{alt}\prime}_{u,v}\sqrt{\mathsf{w}_{u,v}}|u,v\rangle \\
&= \sqrt{\frac{2}{\mathcal{R}^{\mathsf{alt}}_{s,t}}} \sum_{u\in V(G)\setminus\{s\}} |\mathsf{p}^{\mathsf{alt}}_{|u}\rangle.
\end{aligned} \tag{6.9}
$$

In the special case where u has no additional alternative neighbourhoods, for $|\mathsf{p}^{\mathsf{alt}}_{|u}\rangle$ to lie in $\mathrm{span}\{\Psi_\star(u)\} = \mathrm{span}\{|\psi_\star(u)\rangle\}$, the edge potentials $\mathsf{p}_{u,v}$ must be identical for each $v \in \Gamma(u)$, which holds by Ohm's Law.

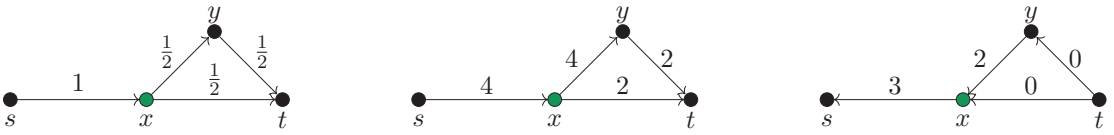
6.2.4 Example graph

Having reestablished the connection between the alternative potential vector and the s-t alternative electrical flow within the multidimensional quantum electrical network



$\mathsf{w}_{u,v}$ for each $(u,v) \in \overrightarrow{E}$ $\theta_{u,v}$ for each $(u,v) \in \overrightarrow{E}$ p_u for each $u \in V$

Figure 6.1: Graph G with its s-t electrical flow θ and corresponding potential p at each vertex.



$\theta^{\mathsf{alt}}_{u,v}$ for each $(u,v) \in \overrightarrow{E}$ $\mathsf{p}^{\mathsf{alt}}_{u,v}$ for each $(u,v) \in \overrightarrow{E}$ $\mathsf{p}^{\mathsf{alt}}_{u,v}$ for each $(v,u) \in \overrightarrow{E}$

Figure 6.2: Graph G where the coloured vertex x has an additional alternative neighbourhood. The s-t alternative electrical flow θ^{alt} with respect to this extra alternative neighbourhood is displayed, as well as the corresponding alternative potential vector $\mathsf{p}^{\mathsf{alt}}$.

framework, we now provide some intuition for these new definitions by revisiting our running example from Figure 3.1. Consider the network $G = (V, E, \mathsf{w})$ with vertex set $V = \{s, x, y, t\}$ and directed edge set $\overrightarrow{E} = \{(s,x), (x,y), (x,t), (y,t)\}$. Each edge $(u,v) \in \overrightarrow{E}$ has a weight $\mathsf{w}_{u,v} = 1/4$, except for the edge (s,x), which has weight $\mathsf{w}_{s,x} = 1$. This network is visualised in Figure 6.1. These directions and weight assignments lead to the following star states for each of the four vertices:

$$|\psi_\star(s)\rangle = |s,x\rangle, \qquad |\psi_\star(x)\rangle = \sqrt{\frac{2}{3}}\left(-|x,s\rangle + \frac{1}{2}|x,y\rangle + \frac{1}{2}|x,t\rangle\right),$$
$$|\psi_\star(y)\rangle = \sqrt{2}\left(-\frac{1}{2}|y,x\rangle + \frac{1}{2}|y,t\rangle\right), \qquad |\psi_\star(t)\rangle = \sqrt{2}\left(-\frac{1}{2}|t,x\rangle - \frac{1}{2}|t,y\rangle\right).$$

In Figure 6.1, we show the s-t electrical flow θ on G along with the corresponding potential vector p. It is straightforward to verify that θ and p satisfy Ohm's Law, meaning $\mathsf{p}_u - \mathsf{p}_v = \frac{\theta_{u,v}}{\mathsf{w}_{u,v}}$ for each edge (u,v).

We now consider the case where the vertex $x \in V$ has an additional alternative neighbourhood. Let $\Psi_\star(x) = \{|\psi_{x,0}\rangle, |\psi_{x,1}\rangle\}$, where:

$$|\psi_{x,1}\rangle = \sqrt{\frac{2}{3}}\left(\frac{1}{2}|x,s\rangle - |x,y\rangle + \frac{1}{2}|x,t\rangle\right),$$

as visualised in Figure 6.2. According to Alternative Kirchhoff's Law, the flow state $|\theta^{\mathsf{alt}}\rangle$ of any unit s-t flow θ^{alt} must also be orthogonal to $|\psi_{x,1}\rangle$. Combined with the condition that θ^{alt} must be orthogonal to all star states, which is equivalent to the flow being conserved at the vertices x and y, this constraint leaves us with a single option for θ^{alt}. The resulting flow is shown in Figure 6.2, and the corresponding flow vector is given by

$$|\theta^{\mathsf{alt}}\rangle = \frac{1}{\sqrt{2\mathcal{R}^{\mathsf{alt}}_{s,t}}} \sum_{(u,v)\in\overrightarrow{E}} \frac{\theta^{\mathsf{alt}}_{u,v}}{\sqrt{\mathsf{w}_{u,v}}} \left(|u,v\rangle + |v,u\rangle\right)$$
$$= \frac{1}{\sqrt{8}}\left(|s,x\rangle + |x,s\rangle + |x,y\rangle + |y,x\rangle + |x,t\rangle + |t,x\rangle + |y,t\rangle + |t,y\rangle\right).$$

Since this θ^{alt} is the only unit s-t flow that satisfies Alternative Kirchhoff's Law, it is, by definition, the s-t alternative electrical flow. To construct the corresponding alternative potential vector $\mathsf{p}^{\mathsf{alt}}$, we start from the states defined in (6.9):

$$\begin{aligned}
|\mathsf{p}^{\mathsf{alt}}_{|s}\rangle &= 4|s,x\rangle,\\
|\mathsf{p}^{\mathsf{alt}}_{|x}\rangle &= -3|x,s\rangle + 2|x,y\rangle + |x,t\rangle,\\
|\mathsf{p}^{\mathsf{alt}}_{|y}\rangle &= -|y,x\rangle + |y,t\rangle,\\
|\mathsf{p}^{\mathsf{alt}}_{|t}\rangle &= 0|t,y\rangle.
\end{aligned}$$

Note that constructing $|\mathsf{p}^{\mathsf{alt}}_{|s}\rangle$ is not strictly necessary, but is included for completeness. Each $|\mathsf{p}^{\mathsf{alt}}_{|u}\rangle$ lies in $\mathrm{span}\{\Psi_\star(u)\}$. The alternative potential vector $\mathsf{p}^{\mathsf{alt}}$ (visualised in Figure 6.2) satisfies $\mathsf{p}^{\mathsf{alt}}_{s,x} = \mathcal{R}^{\mathsf{alt}}_{s,t} = 4$ and $\mathsf{p}^{\mathsf{alt}}_{t,x} = \mathsf{p}^{\mathsf{alt}}_{t,y} = 0$, as well as Alternative Ohm's Law, ensuring that for each $(u,v) \in E$, we have $\mathsf{p}^{\mathsf{alt}}_{u,v} - \mathsf{p}^{\mathsf{alt}}_{v,u} = \theta^{\mathsf{alt}}_{u,v}/\mathsf{w}_{u,v}$. Thus, we have found the alternative potential vector $\mathsf{p}^{\mathsf{alt}}$ whose associated state $|\mathsf{p}^{\mathsf{alt}}\rangle$ satisfies $\Pi_{\mathcal{A}^{\mathsf{alt}}}|\mathsf{p}^{\mathsf{alt}}\rangle = |\mathsf{p}^{\mathsf{alt}}\rangle$:

$$\begin{aligned}
|\mathsf{p}^{\mathsf{alt}}\rangle &= \sqrt{\frac{2}{\mathcal{R}^{\mathsf{alt}}_{s,t}}} \sum_{u\in V\setminus\{s\}} \sum_{v\in\Gamma(u)} (-1)^{\Delta_{u,v}} \mathsf{p}^{\mathsf{alt}}_{u,v} \sqrt{\mathsf{w}_{u,v}}|u,v\rangle\\
&= -3|x,s\rangle + 2|x,y\rangle + |x,t\rangle - |y,x\rangle + |y,t\rangle\\
&= -\sqrt{\frac{3}{2}}\left(\frac{8}{3}|\psi_x\rangle + \frac{2}{3}|\psi^{\mathsf{alt}}_x\rangle\right) + \sqrt{2}|\psi_y\rangle + 0|\psi_t\rangle.
\end{aligned}$$

6.2.5 The alternative incidence matrix

In this section, inspired by the connection between any electrical network $G = (V, E, \mathsf{w})$ and its incidence matrix B from Section 3.2.2, we extend this connection to the multidimensional electrical network and its corresponding alternative incidence matrix B_{alt}. We will use this connection to prove the uniqueness of the s-t alternative flow θ^{alt} and the existence of the alternative potential $\mathsf{p}^{\mathsf{alt}}$, which together satisfy Alternative Ohm's Law.

What makes the s-t electrical flow θ special, such that it satisfies Ohm's Law? And why does Ohm's Law not necessarily hold for the s-t alternative flow? Although all flow states lie in the symmetric subspace $\mathcal{B}^\perp$ by construction (see (3.29)), we saw in (3.43) that the flow state $|\theta\rangle$ of the s-t electrical flow θ can be written as

$$|\theta\rangle = (I - \Pi_{\mathcal{B}})\sqrt{\frac{2}{\mathcal{R}_{s,t}}} \sum_{u\in V} \mathsf{p}_u \sqrt{\mathsf{w}_u}|\psi_\star(u)\rangle,$$

indicating that $|\theta\rangle$ actually resides in the *symmetric star space* of $\mathcal{H}$, which is a subspace of $\mathcal{B}^\perp$:

$$H^{+\star} := \mathrm{span}\{(I - \Pi_{\mathcal{B}})|\psi_\star(u)\rangle : u \in V\}.$$

For the rest of this section, we will abbreviate the normalised projection of any star state $|\psi_\star(u)\rangle$ onto $\mathcal{B}^\perp$ as

$$|\psi_\star(u)^+\rangle := \sqrt{2}(I - \Pi_{\mathcal{B}})|\psi_\star(u)\rangle = \frac{I + \mathsf{SWAP}}{\sqrt{2}}|\psi_\star(u)\rangle.$$

Among all s-t flows, the s-t electrical flow is the unique unit flow for which the corresponding flow state $|\theta\rangle$ lies entirely in $H^{+\star}$ (see e.g., [LP16]). While we do not provide a formal proof of this statement, the intuition is that any other s-t flow has higher energy,

i.e. a higher norm, due to containing a component that is orthogonal to $H^{+\star}$: a circulation. The column space of the incidence matrix B is isomorphic to $H^{+\star}$, where each column indexed by $u \in V$ represents $\sqrt{\mathsf{w}_u}|\psi_\star(u)^+\rangle$ through the isometry

$$\mathcal{V} : \mathbb{C}^{|\overrightarrow{E}|} \mapsto \mathcal{B}^\perp, \text{ where } \mathcal{V}(u,v) = \sqrt{2}(I - \Pi_{\mathcal{B}})|u,v\rangle = \frac{1}{\sqrt{2}}\left(|u,v\rangle + |v,u\rangle\right). \tag{6.10}$$

By introducing alternative neighbourhoods (see Definition 4.2.1), we effectively enlarge the space $H_G^{+\star}$. We define

$$V^{\mathsf{alt}} := \{(u,i) \in V \times \mathbb{N}_{\geq 0} : i \in [a_u - 1]_0\}.$$

Instead of only considering the span of all $|\psi_\star(u)^+\rangle$ for $u \in V$, we now consider the span of all alternative neighbourhoods projected onto the symmetric subspace. For each $(u,i) \in V^{\mathsf{alt}}$, we define $|\overline{\psi}^+_{u,i}\rangle := \sqrt{2}(I - \Pi_{\mathcal{B}})|\overline{\psi}_{u,i}\rangle$, leading to the expanded space

$$H^{+\mathsf{alt}} := \operatorname{span}\{|\overline{\psi}^+_{u,i}\rangle : u \in V, i \in [a_u - 1]_0\}.$$

By modifying the incidence matrix B to ensure that its column space represents the enlarged space $H_G^{+\mathsf{alt}}$, we obtain the alternative incidence matrix B_{alt}:

Definition 6.2.5 (Alternative incidence matrix)**.** *Let G be a network, and let $\Psi_\star$ be a collection of alternative neighbourhoods. For each $\Psi_\star(u) \in \Psi_\star$, let $\{|\overline{\psi}_{u,0}\rangle, \ldots, |\overline{\psi}_{u,a_u-1}\rangle\}$ form an orthonormal basis for $\Psi_\star(u)$. The alternative incidence matrix $B_{\mathsf{alt}} \in \mathbb{C}^{\overrightarrow{E} \times V^{\mathsf{alt}}}$ of G is a matrix whose rows are indexed by $(u,v) \in \overrightarrow{E}$, whose columns are indexed by $(u,i) \in V^{\mathsf{alt}}$, and whose non-zero entries are given by*

$$B_{\mathsf{alt}(u,v),(u,i)} = \sqrt{\mathsf{w}_u}\langle u,v|\overline{\psi}_{u,i}\rangle, \quad B_{\mathsf{alt}(u,v),(v,j)} = \sqrt{\mathsf{w}_v}\langle u,v|\overline{\psi}_{v,j}\rangle.$$

By Definition 4.2.1, we may assume $|\overline{\psi}_{u,0}\rangle = |\psi_\star(u)\rangle$. Substituting B with B_{alt} in (3.2) and (3.4) allows us to recover both Alternative Kirchhoff's Law and Alternative Ohm's Law, confirming that these are in fact natural definitions from the perspective of the incidence matrix. Let us fix an ordering of the columns of B as $s, (u_1,i_1), \ldots, (u_2,i_2), t$, where $(u_1,i_1), (u_2,i_2) \in V^{\mathsf{alt}}$ and $u_1, u_2 \in V \backslash \{s,t\}$.

Definition 6.2.6 (Alternative Kirchhoff's Law (incidence matrix))**.** *Let θ^{alt} be an alternative unit s-t flow on an electrical network $G = (V,E,\mathsf{w})$ with respect to a collection of alternative neighbourhoods $\Psi_\star$. Let B_{alt} be the alternative incidence matrix of G. Then θ^{alt} satisfies:*

$$B_{\mathsf{alt}}^T W \theta^{\mathsf{alt}} = \begin{bmatrix} \sum_{v \in \Gamma(s)} \theta^{\mathsf{alt}}_{s,v} \\ \sum_{v \in \Gamma(u_1)} \frac{\theta^{\mathsf{alt}}_{u_1,v}}{\sqrt{\mathsf{w}_{u_1,v}}} \sqrt{\mathsf{w}_{u_1}} \langle u_1, v|\overline{\psi}_{u_1,i_1}\rangle \\ \vdots \\ \sum_{v \in \Gamma(u_2)} \frac{\theta^{\mathsf{alt}}_{u_2,v}}{\sqrt{\mathsf{w}_{u_2,v}}} \sqrt{\mathsf{w}_{u_2}} \langle u_2, v|\overline{\psi}_{u_2,i_2}\rangle \\ \sum_{v \in \Gamma(t)} \theta^{\mathsf{alt}}_{t,v} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix} = \mathsf{e}_s - \mathsf{e}_t. \tag{6.11}$$

Recall from Definition 6.2.2 that the s-t alternative electrical flow is the flow that minimises $\mathcal{E}(\theta^{\mathsf{alt}})$ among all alternative unit s-t flows (if any such flow exists). By applying the Moore-Penrose inverse of B_{alt}, denoted $B^{\dagger+}_{\mathsf{alt}}$ (to avoid double superscripts), to (6.11), we can prove the uniqueness of the s-t alternative electrical flow:

Theorem 6.2.7. *Let θ^{alt} be the s-t alternative electrical flow on a network $G = (V,E,\mathsf{w})$, and let B_{alt} be the alternative incidence matrix of G. Then $W\theta^{\mathsf{alt}}$ is given by*

$$W\theta^{\mathsf{alt}} = B^{\dagger+}_{\mathsf{alt}}(\mathsf{e}_s - \mathsf{e}_t). \tag{6.12}$$

Recall the isometry $\mathcal{V}$ from (6.10). The column of B_{alt} indexed by $(u,i) \in V^{\mathsf{alt}}$ is equal to $\mathcal{V}^\dagger\left(\sqrt{\mathsf{w}_u}|\overline{\psi}^+_{u,i}\rangle\right)$, implying that the column space of B_{alt} is equal to $\mathcal{V}^\dagger\left(H^{+\mathsf{alt}}\right)$. Furthermore, the column space of B_{alt} is also equal to the column space of $B^{\dagger+}_{\mathsf{alt}}$ due to the properties of the Moore-Penrose inverse (see (2.2)). Given that the state $|\theta^{\mathsf{alt}}\rangle$ is related to the vector $W\theta^{\mathsf{alt}}$ via $\sqrt{\mathcal{R}^{\mathsf{alt}}_{s,t}}|\theta^{\mathsf{alt}}\rangle = \mathcal{V}\left(W\theta\right)$, it follows that $|\theta^{\mathsf{alt}}\rangle$ is an element of $H^{+\mathsf{alt}}$. Thus, there exist coefficients $\mathsf{p}^{\mathsf{alt}}_{(u,i)}$ such that:

$$|\theta^{\mathsf{alt}}\rangle = \frac{1}{\sqrt{\mathcal{R}^{\mathsf{alt}}_{s,t}}}\sum_{u\in V}\sum_{i=0}^{a_u-1}\mathsf{p}^{\mathsf{alt}}_{(u,i)}\sqrt{\mathsf{w}_u}|\overline{\psi}^+_{u,i}\rangle. \tag{6.13}$$

The notation $\mathsf{p}^{\mathsf{alt}}_{(u,i)}$ suggests that these coefficients are related to the alternative potential vector $\mathsf{p}^{\mathsf{alt}}$. This is indeed the case: by defining the potential vector $\mathsf{p}^{\mathsf{alt}}$ as

$$\mathsf{p}^{\mathsf{alt}}_{u,v} := \frac{(-1)^{\Delta_{u,v}}}{\sqrt{\mathsf{w}_{u,v}}}\sum_{i=0}^{a_u-1}\mathsf{p}^{\mathsf{alt}}_{(u,i)}\sqrt{\mathsf{w}_u}\langle u,v|\overline{\psi}_{u,i}\rangle, \tag{6.14}$$

we ensure that the state $|\mathsf{p}^{\mathsf{alt}}\rangle$ satisfies $|\mathsf{p}^{\mathsf{alt}}\rangle \in \mathcal{A}^{\mathsf{alt}}$:

$$\begin{aligned}
|\mathsf{p}^{\mathsf{alt}}\rangle &= \sqrt{\frac{2}{\mathcal{R}^{\mathsf{alt}}_{s,t}}}\sum_{u\in V\setminus\{s\}}\sum_{v\in\Gamma(u)}(-1)^{\Delta_{u,v}}\mathsf{p}^{\mathsf{alt}}_{u,v}\sqrt{\mathsf{w}_{u,v}}|u,v\rangle \\
&= \sqrt{\frac{2}{\mathcal{R}^{\mathsf{alt}}_{s,t}}}\sum_{u\in V\setminus\{s\}}\sum_{v\in\Gamma(u)}\sum_{i=0}^{a_u-1}\mathsf{p}^{\mathsf{alt}}_{(u,i)}\sqrt{\mathsf{w}_u}\langle u,v|\overline{\psi}_{u,i}\rangle|u,v\rangle \\
&= \sqrt{\frac{2}{\mathcal{R}^{\mathsf{alt}}_{s,t}}}\sum_{u\in V\setminus\{s\}}\sum_{i=0}^{a_u-1}\mathsf{p}^{\mathsf{alt}}_{(u,i)}\sqrt{\mathsf{w}_u}|\overline{\psi}_{u,i}\rangle.
\end{aligned} \tag{6.15}$$

Note that for any vertex $u \in V$, if u does not have any additional alternative neighbourhoods (this is particularly true for $s,t \in V$), then by (6.14), we have that $\mathsf{p}^{\mathsf{alt}}_{u,v} = \mathsf{p}^{\mathsf{alt}}_{(u,0)}$ for any neighbor $v \in \Gamma(u)$.

Due to the coefficients $\mathsf{p}^{\mathsf{alt}}_{(u,i)}$, we can view the alternative potential vector $\mathsf{p}^{\mathsf{alt}}$ as a vector in $\mathbb{C}^{V^{\mathsf{alt}}}$, with entries $\mathsf{p}^{\mathsf{alt}}_{(u,i)}$ for the row indexed by $(u,i) \in V^{\mathsf{alt}}$. By substituting B with B_{alt} in (3.4) and using (6.14), we recover Alternative Ohm's Law:

Definition 6.2.8 (Alternative Ohm's Law (incidence matrix))**.** *Let θ^{alt} be any alternative unit s-t flow on an electrical network $G = (V,E,\mathsf{w})$ with respect to a collection of alternative neighbourhoods $\Psi_\star$. Let B_{alt} be the alternative incidence matrix of G. Then the alternative potential vector $\mathsf{p}^{\mathsf{alt}}$ is an alternative potential vector such that $\Pi_{\mathcal{A}^{\mathsf{alt}}}|\mathsf{p}^{\mathsf{alt}}\rangle = |\mathsf{p}^{\mathsf{alt}}\rangle$, $\mathsf{p}^{\mathsf{alt}}_s = \mathcal{R}^{\mathsf{alt}}_{s,t}$, $\mathsf{p}^{\mathsf{alt}}_t = 0$ and*

$$B_{\mathsf{alt}}\mathsf{p}^{\mathsf{alt}} = \begin{bmatrix}\sqrt{\mathsf{w}_{u_1,v_1}}\left(\mathsf{p}^{\mathsf{alt}}_{u_1,v_1} - \mathsf{p}^{\mathsf{alt}}_{v_1,u_1}\right)\\ \vdots \\ \sqrt{\mathsf{w}_{u_2,v_2}}\left(\mathsf{p}^{\mathsf{alt}}_{u_2,v_2} - \mathsf{p}^{\mathsf{alt}}_{v_2,u_2}\right)\end{bmatrix} = \begin{bmatrix}\frac{\theta_{u_1,v_1}}{\sqrt{\mathsf{w}_{u_1,v_1}}}\\ \vdots \\ \frac{\theta_{u_2,v_2}}{\sqrt{\mathsf{w}_{u_2,v_2}}}\end{bmatrix} = W\theta^{\mathsf{alt}}. \tag{6.16}$$

Just like with the potential vector p, we may assume that the alternative potential vector $\mathsf{p}^{\mathsf{alt}}$ satisfying Alternative Ohm's Law also satisfies $\mathsf{p}^{\mathsf{alt}}_s = \mathcal{R}^{\mathsf{alt}}_{s,t}$ and $\mathsf{p}^{\mathsf{alt}}_t = 0$.

Theorem 6.2.9. *Let θ^{alt} be the s-t alternative electrical flow on an electrical network $G = (V,E,\mathsf{w})$ with respect to a collection of alternative neighbourhoods $\Psi_\star$. Let B_{alt} be the alternative incidence matrix of G. Then there exists an alternative potential vector $\mathsf{p}^{\mathsf{alt}}$ satisfying Alternative Ohm's Law.*

Proof. Recall from Lemma 3.2.12 that B has $\sum_{u\in V}\mathsf{e}_u$ in its kernel. It is therefore straightforward to see that $\sum_{u\in V}\mathsf{e}_{u,0}$ lives in the kernel of B_{alt}. This allows us to apply the same trick as in (3.5), meaning we remove the last column of B_{alt} and last row of $\mathsf{p}^{\mathsf{alt}}$ to obtain $\overline{B_{\mathsf{alt}}}$ and $\overline{\mathsf{p}^{\mathsf{alt}}}$, forcing $\mathsf{p}^{\mathsf{alt}}_t = 0$ for the solution satisfying (6.16):

$$\mathsf{p}^{\mathsf{alt}} = \begin{bmatrix}\overline{\mathsf{p}^{\mathsf{alt}}}\\ 0\end{bmatrix} = \begin{bmatrix}\overline{B_{\mathsf{alt}}}^{+} W\theta^{\mathsf{alt}}\\ 0\end{bmatrix}. \tag{6.17}$$

By left-multiplying both sides of (6.12) with $(W\theta^{\mathsf{alt}})^T$ we obtain together with (6.16) that

$$\mathcal{R}^{\mathsf{alt}}_{s,t} = \left\|W\theta^{\mathsf{alt}}\right\|^2 = (W\theta^{\mathsf{alt}})^T B^{\dagger+}_{\mathsf{alt}}(\mathsf{e}_s-\mathsf{e}_t) = \mathsf{p}^{\mathsf{alt}\,T}(\mathsf{e}_s-\mathsf{e}_t) = \mathsf{p}^{\mathsf{alt}}_s - \mathsf{p}^{\mathsf{alt}}_t = \mathsf{p}^{\mathsf{alt}}_s. \tag{6.18}$$

□

Due to Theorem 6.2.9 and (6.8), we may now apply Lemma 3.3.11 with the parameters $|\psi_0\rangle = \sqrt{2}(I-\Pi_{\mathcal{B}})|\psi_\star(s_0)\rangle$, $|w^{\mathsf{alt}}\rangle = \sqrt{2}|\theta^{\mathsf{alt}\prime}\rangle$ and $|w^{\mathsf{alt}}_{\mathcal{A}}\rangle = \frac{1}{\sqrt{2}}|\mathsf{p}^{\mathsf{alt}\prime}\rangle$ (see (3.28), (3.34) and (6.7), respectively), proving the generalisation of Corollary 3.4.2 to alternative neighbourhoods. Here we generalise the notion of escape time ET_s from (3.42) using the decomposition of our alternative potential vector from (6.15):

$$\mathsf{ET}^{\mathsf{alt}}_s := \frac{1}{\mathcal{R}^{\mathsf{alt}}_{s,t}}\sum_{u\in V(G)}\sum_{i=0}^{a_u-1}\left(\mathsf{p}^{\mathsf{alt}}_{(u,i)}\right)^2\mathsf{w}_u, \tag{6.19}$$

such that we can bound the norm of $|\mathsf{p}^{\mathsf{alt}\prime}\rangle$:

$$\begin{aligned}\left\||\mathsf{p}^{\mathsf{alt}\prime}\rangle\right\|^2 &= \frac{1}{\mathcal{R}^{\mathsf{alt}}_{s,t}}\sum_{u\in V(G)}\sum_{i=0}^{a_u-1}\left(\mathsf{p}^{\mathsf{alt}}_{(u,i)}\right)^2\mathsf{w}^{G'}_u\\ &= \frac{1}{\mathcal{R}^{\mathsf{alt}}_{s,t}}\sum_{u\in V(G)}\sum_{i=0}^{a_u-1}\left(\mathsf{p}^{\mathsf{alt}}_{(u,i)}\right)^2\mathsf{w}_u + \mathcal{R}^{\mathsf{alt}}_{s,t}\mathsf{w}_0 = \mathsf{ET}^{\mathsf{alt}}_s + 1.\end{aligned}$$

Theorem 6.2.10. *Fix a network $G=(V,E,\mathsf{w})$ as in Definition 3.2.1 with vertices $s,t\in V$ and let $\Psi_\star$ be a collection of alternative neighbourhoods on G. Let $U_{\mathcal{A}^{\mathsf{alt}}\mathcal{B}}$ be the quantum walk operator with respect to $\Psi_\star$ as defined in (4.1). Let θ^{alt} be the s-t electrical flow on G with corresponding flow state $|\theta^{\mathsf{alt}}\rangle$ as defined in (6.2). If θ^{alt} exists, then by performing $T=(17\pi^2/16\sqrt{2}\epsilon^2)\sqrt{\mathsf{ET}^{\mathsf{alt}}_s+1}$ steps of phase estimation on the initial state $|\psi_0\rangle$ as defined in (3.28) with the operator $U_{\mathcal{A}^{\mathsf{alt}}\mathcal{B}}$, the phase estimation algorithm outputs "0" with bounded error, leaving a state $|\widetilde{\theta}\rangle$ satisfying*

$$\frac{1}{2}\left\||\widetilde{\theta}\rangle\langle\widetilde{\theta}| - |\theta^{\mathsf{alt}}\rangle\langle\theta^{\mathsf{alt}}|\right\|_1 \le \epsilon.$$

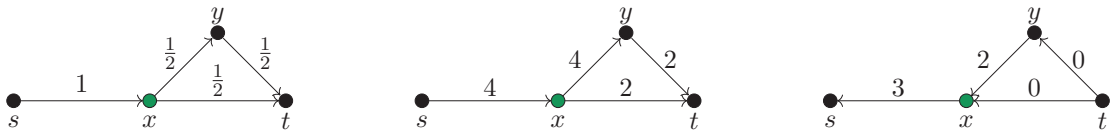
6.2.6 Example graph

We now demonstrate how these results apply to the example from Figure 6.2, which we have restated here in Figure 6.3. We consider the case where the vertex $x\in V$ contains an additional alternative neighbourhood: let $\Psi_\star(x)=\{|\psi_x\rangle,|\psi^{\mathsf{alt}}_x\rangle\}$, where

$$|\psi^{\mathsf{alt}}_x\rangle = \sqrt{\frac{2}{3}}\left(\frac{1}{2}|x,s\rangle - |x,y\rangle + \frac{1}{2}|x,t\rangle\right).$$

By using

$$\sqrt{\mathsf{w}_x}|\psi_{x,1}\rangle = \sqrt{\frac{3}{2}}\sqrt{\frac{1}{2}}\left(-|x,y\rangle+|x,t\rangle\right) = \frac{1}{2}\sqrt{3}\left(-|x,y\rangle+|x,t\rangle\right),$$



$\theta^{\mathsf{alt}}_{u,v}$ for each $(u,v) \in \overrightarrow{E}$ $\quad$ $\mathsf{p}^{\mathsf{alt}}_{u,v}$ for each $(u,v) \in \overrightarrow{E}$ $\quad$ $\mathsf{p}^{\mathsf{alt}}_{u,v}$ for each $(v,u) \in \overrightarrow{E}$

Figure 6.3: Graph G where the coloured vertex x has an additional alternative neighbourhood. The s-t alternative electrical flow θ^{alt} with respect to this extra alternative neighbourhood is displayed, as well as the corresponding alternative potential vector $\mathsf{p}^{\mathsf{alt}}$.

we find that $\{|\psi_{x,0}\rangle = |\psi_x\rangle, |\psi_{x,1}\rangle\}$ forms an orthonormal basis for $\Psi_\star(x)$. For this basis, the alternative incidence matrix B_{alt} of G and $\Psi_\star$, along with its Moore-Penrose inverse $B^{\dagger+}_{\mathsf{alt}}$, is given by

$$B_{\mathsf{alt}} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & -\frac{1}{2}\sqrt{3} & -\frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2}\sqrt{3} & 0 & -\frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & -\frac{1}{2} \end{bmatrix}, \quad B^{\dagger+}_{\mathsf{alt}} = \begin{bmatrix} \frac{3}{4} & -\frac{1}{4} & 0 & -\frac{1}{4} & -\frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{3}\sqrt{3} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{3}\sqrt{3} & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{3}\sqrt{3} & 1 & -1 \end{bmatrix}. \tag{6.20}$$

We can now recover the electrical flow θ^{alt} with respect to $\Psi_\star$, as shown in Figure 6.3, using Theorem 6.2.7 to derive

$$W\theta^{\mathsf{alt}} = \begin{bmatrix} \frac{\theta^{\mathsf{alt}}_{s,x}}{\sqrt{\mathsf{w}_{s,x}}} \\ \frac{\theta^{\mathsf{alt}}_{x,y}}{\sqrt{\mathsf{w}_{x,y}}} \\ \frac{\theta^{\mathsf{alt}}_{x,t}}{\sqrt{\mathsf{w}_{x,t}}} \\ \frac{\theta^{\mathsf{alt}}_{y,t}}{\sqrt{\mathsf{w}_{y,t}}} \end{bmatrix} = B^{\dagger+}_{\mathsf{alt}}(\mathsf{e}_s - \mathsf{e}_t) = \begin{bmatrix} \frac{3}{4} & -\frac{1}{4} & 0 & -\frac{1}{4} & -\frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{3}\sqrt{3} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{3}\sqrt{3} & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & -\frac{1}{3}\sqrt{3} & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Thus, $\mathcal{R}^{\mathsf{alt}}_{s,t} = 1+1+1+1 = 4$. Using (6.17), the matrix $\overline{B_{\mathsf{alt}}}$ and its Moore-Penrose inverse $\overline{B_{\mathsf{alt}}}^{+}$ are:

$$\overline{B_{\mathsf{alt}}} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & \frac{1}{2} & -\frac{1}{2}\sqrt{3} & -\frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2}\sqrt{3} & 0 \\ 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}, \quad \overline{B_{\mathsf{alt}}}^{+} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & -\frac{1}{2}\sqrt{3} & \frac{1}{2}\sqrt{3} & -\frac{1}{2}\sqrt{3} \\ 0 & 0 & 0 & 2 \end{bmatrix}, \tag{6.21}$$

The alternative potential at each alternative neighbourhood is then given by

$$\mathsf{p}^{\mathsf{alt}} = \begin{bmatrix} \overline{\mathsf{p}^{\mathsf{alt}}} \\ 0 \end{bmatrix} = \begin{bmatrix} \overline{B_{\mathsf{alt}}}^{+} W \theta^{\mathsf{alt}}_{s,t} \\ 0 \end{bmatrix} = \left[\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & -\frac{1}{3}\sqrt{3} & \frac{1}{3}\sqrt{3} & -\frac{1}{3}\sqrt{3} \\ 0 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \right] = \begin{bmatrix} 4 \\ 3 \\ -\frac{1}{3}\sqrt{3} \\ 2 \\ 0 \end{bmatrix}. \tag{6.22}$$

This confirms that $\mathsf{p}^{\mathsf{alt}}_{(s,0)} = \mathsf{p}^{\mathsf{alt}}_{s,x} = \mathcal{R}^{\mathsf{alt}}_{s,t} = 4$, and the resulting potential state $|\mathsf{p}^{\mathsf{alt}}\rangle$ is equal to

$$\begin{aligned} |\mathsf{p}^{\mathsf{alt}}\rangle &= \sqrt{\frac{2}{\mathcal{R}^{\mathsf{alt}}_{s,t}}} \sum_{u \in V(G)\setminus\{s\}} \sum_{i=0}^{a_u - 1} \mathsf{p}^{\mathsf{alt}}_{(u,i)} \sqrt{\mathsf{w}_u} |\psi_{(u,i)}\rangle \\ &= -3|x,s\rangle + \frac{3}{2}|x,y\rangle + \frac{3}{2}|x,t\rangle + \frac{1}{2}|x,y\rangle - \frac{1}{2}|x,t\rangle - |y,x\rangle + |y,t\rangle \\ &= -3|x,s\rangle + 2|x,y\rangle + |x,t\rangle - |y,x\rangle + |y,t\rangle. \end{aligned} \tag{6.23}$$

6.3 Electrical flow sampling on one-dimensional random hierarchical graphs

Recently, [BLH23] have shown that there is an exponential separation between quantum and classical algorithms in finding a marked vertex in one-dimensional random hierarchical graphs, which is a generalisation of the result of the welded trees problem [CCD+03]. In this section, we show that for one-dimensional random hierarchical graphs, we can efficiently generate a set of alternative neighbourhoods $\Psi_\star$ such that the resulting s-t alternative electrical flow matches the s-t electrical flow, meaning it satisfies Ohm's Law. We then invoke Theorem 6.2.10, allowing us to efficiently approximate the s-t electrical flow and sample from it to find a marked vertex, recovering some of the results from [BLH23].

6.3.1 One-dimensional random hierarchical graphs

Following [BLH23], we now define the one-dimensional random hierarchical graph model with nodes $S_0, \ldots, S_n$.

Definition 6.3.1 (Hierarchical graph on a line supergraph $\mathcal{G}$)**.** *A hierarchical graph on a line supergraph $\mathcal{G} = (\mathcal{V} = [n]_0, \mathcal{E})$ of length n is defined by a set of nodes S_v for each $v \in \mathcal{V}$ and a set of edges $E_{u,v}$ for each $(u,v) \in \mathcal{E}$ such that $s_v = |S_v|$ and $e_{(u,v)} = |E_{u,v}|$. There are two special start and exit nodes $S_0 = \{s\}$ and $S_n = \{t\}$, meaning $s_0 = s_n = 1$. Define $V = \bigcup_{v\in\mathcal{V}} S_v, E = \bigcup_{(u,v)\in\mathcal{E}} E_{u,v}$ and $G = (V, E)$. For each $(u,v) \in E(G)$, the edge set $E_{u,v}$ denotes the set of edges between the nodes between S_u and S_v.*

Here, the term "one-dimensional" refers to the fact that the supergraph $\mathcal{G}$ forms a line. In Section 6.3.4, we revisit the welded trees graph from Section 4.4, which serves as an example of a one-dimensional random hierarchical graph. For additional examples, we refer the reader to the original work in [BLH23]. We restrict our attention to a subclass of the above definition such that the hierarchical graph exhibits enough symmetry for the quantum algorithm to efficiently explore the graph:

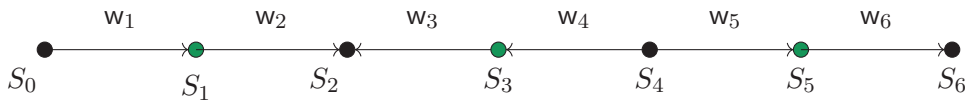


Figure 6.4: A line supergraph $\mathcal{G}$ with nodes $S_0, \ldots, S_6$. The black nodes are subsets of V_{even}, where the edge directions are reversed and where all adjacent edges have the same weight and direction.

Definition 6.3.2 (Balanced hierarchical graph)**.** *A hierarchical graph on a supergraph G is said to be* balanced *if for every $(u, v) \in E(G)$, the number of edges connecting a fixed node $\alpha \in S_u$ to nodes in S_v is the same for each α.*

Definition 6.3.3 (Edge-edge ratio)**.** *Consider a hierarchical graph on the line supergraph G with nodes $S_0, \ldots, S_n$, where each node S_i contains $s_0, \ldots, s_n$ vertices. Let e_k and $\mathcal{E}_k$ denote the number of edges and the set of edges between the nodes S_{k-1} and S_k respectively. Then the* edge ratios *r_k for $k \in [n-1]$ are defined as*

$$r_k = \frac{e_{k+1}}{e_k}.$$

Definition 6.3.4 (Edge-vertex ratio)**.** *A hierarchical graph on the line supergraph $G = (V, E)$ with nodes $S_0, \ldots, S_n$ possesses* edge-vertex ratios *$\kappa_1, \ldots, \kappa_n$ given by*

$$\kappa_j = \frac{e_j}{s_j}. \tag{6.24}$$

Lastly, we require that the above graph be D-regular. This ensures that the degree of a vertex does not reveal any information about its position in the graph. Note that when we refer to a graph as being D-regular, as is the case for the welded trees graph from Section 4.4, we allow for the entrance and exit vertices to have a different degree. For the rest of this section, we assume that the these vertices are of degree $D-1$. For a D-regular random balanced hierarchical graph on a line supergraph $G = (V, E)$, we have

$$e_i + e_{i+1} = e_i + r_i e_i = \kappa_i s_i + r_i \kappa_i s_i = D s_i, \qquad \kappa_i(1 + r_i) = D. \tag{6.25}$$

Let $\ell = \Theta(n)$ be an integer such that $2^\ell \gg |V|$, where $|V|$ is the number of vertices in the one-dimensional random hierarchical graph G. This imposes the restriction that $|V|$ can be at most exponential in n. To each vertex in V, we assign a random name from the set $\{0,1\}^\ell$. To access the neighbours of a particular vertex, we are given quantum access to an adjacency list oracle O_G for the graph G. Given an ℓ-bit string $\sigma \in \{0,1\}^\ell$ corresponding to a vertex $u \in V$, the adjacency list oracle O_G provides the bit strings of the neighbouring vertices in $\Gamma(u)$. If σ does not correspond to any vertex, which will often be the case since $2^\ell \gg |V|$, the oracle instead returns $\perp$. This oracle structure effectively forces any algorithm to start at s and traverse the graph G from there, as it is infeasible to try and guess the name of any other vertex in V.

Problem 6.3.5 (One-dimensional random hierarchical graph problem)**.** *We are given an adjacency list oracle O_G to the one-dimensional random hierarchical graph G (a D-regular graph) on the line supergraph of length n, and the possibility to check whether any vertex u is equal to t. Given the ℓ-bit string associated with the starting vertex $s \in \{0,1\}^\ell$, the goal is to output the ℓ-bit string corresponding to the other root t.*

6.3.2 The electrical network

Before we can use Theorem 6.2.10 to tackle this problem, we must turn G into an electrical network (see Definition 3.2.2), meaning we need to assign a weight and direction to each

of its edges. We assign all edges in $E_k = \bigcup_{(u,v)\in\mathcal{E}_k} E_{u,v}$ the same weight for $k \in [n]$, and the weight w_k changes every two layers (starting at w_2). Without loss of generality, we assume that n is an even number and set $\mathsf{w}_1 = 1$ and

$$\mathsf{w}_k = \prod_{i=1}^{\lfloor k/2 \rfloor} \left(\frac{1}{r_{2i-1}}\right)^2. \tag{6.26}$$

For each vertex $u \in S_i$ where $i \in [n-1]$, we find that

$$\mathsf{w}_u = \sum_{v\in\Gamma(u)} \mathsf{w}_{u,v} = \kappa_i \mathsf{w}_i + (D - \kappa_i)\mathsf{w}_{i+1}.$$

We define the set of directed edges as follows:

$$\overrightarrow{E} = \bigcup_{k \bmod 4 \in \{1,2\}} \{(u,v) : u \in S_{k-1}, v \in S_k\} \cup \bigcup_{k \bmod 4 \in \{0,3\}} \{(u,v) : v \in S_{k-1}, u \in S_k\}. \tag{6.27}$$

See Figure 6.4 for an example of a line supergraph where this edge orientation and weight assignments are visualised. By viewing G as an electrical network, it is straightforward to compute the effective resistance $\mathcal{R}_{s,t}$ via the resistance laws for electrical circuits in series and parallel [Sie86]. As a result, we find for the weight assignment from (6.26) that

$$\mathcal{R}_{s,t} = \sum_{k=1}^{n} \frac{1}{e_k \mathsf{w}_k}. \tag{6.28}$$

We can verify this by noting that this effective resistance is obtained via the natural choice of flow θ that, for each vertex u in the node S_i for $i \in [n-1]$, comes in from the parent in node S_{i-1} and then distributes this flow evenly along all of each children in node S_{i+1}. That is,

$$\forall k \in [n], e \in E_k, \quad \theta_e := \frac{1}{e_k}. \tag{6.29}$$

Recall that we defined the weights in (6.26) and the edge directions in (6.27) in an alternating fashion. This induces a partition of V into $V_{\text{even}} = \bigcup_{v\in\mathcal{V}: v \text{ is even}} S_v$ and $V_{\text{odd}} = \bigcup_{v\in\mathcal{V}: v \text{ is odd}} S_v$. Recall from Section 4.4.8 that we can assume without loss of generality that we know for any $u \in V$ whether it belongs to V_{even} or V_{odd} by keeping track of the parity of the distance from s, which is initially 0 and flips every time the algorithm takes a step.

At each vertex in V_{even} the edge directions are reversed and all adjacent edges have the same weight (see Figure 6.4). It is therefore straightforward to generate the star state $|\psi_\star(v)\rangle$ for each $v \in V_{\text{even}} \setminus \{s,t\}$, since:

$$|\psi_\star(v)\rangle = \pm\frac{1}{\sqrt{D}} \left(|v,u_1\rangle + \cdots + |v,u_D\rangle\right). \tag{6.30}$$

So even though we don't know which supernode v is in, nor which of it neighbours are in S_{i-1} or S_{i+1}, we can always generate (6.30) up to a sign difference, which poses no problem if we wish to reflect around its span. To avoid possible confusion, we emphasise that in Section 4.4 this was the case for V_{odd}, due to a slightly different weight and edge direction assignment.

On the other hand, for each $u \in V_{\text{odd}} \setminus \{t\}$, the situation is more complicated. Let k such that $u \in S_k$ and let $v_1, \ldots, v_\ell \in \Gamma(u) \cap S_{k-1}$ be the neighbours of u that lay in the

node S_{k-1} and similarly let $v_{\ell+1}, \dots, v_D \in \Gamma(u) \cap S_{k+1}$ be the neighbours of u that lay in the node S_{k+1}, where $\ell = \frac{D}{1+r_k}$, which is in fact an integer. Then

$$|\psi_\star(u)\rangle \propto \pm \sum_{i=1}^{\ell} -e_{k+1}|u, v_i\rangle + \sum_{i=\ell+1}^{D} e_k|u, v_i\rangle. \tag{6.31}$$

Although we can learn all the neighbours of u with a single query, we can't distinguish which neighbours are in S_{i-1} or S_{i+1} and it is hence computationally infeasible to generate (6.31). This problem can be circumvented using the alternative neighbourhood technique (see Definition 4.2.1) and since one-dimensional random hierarchical graphs generalise the welded trees graph, it should come as no surprise that we will use a collection of alternative neighbourhoods that generalises the one used in Claim 4.4.4 to traverse the welded trees graph:

Definition 6.3.6 (Alternative Fourier Neighbourhood)**.** *For any positive integer D, let $\omega_D = \exp(2\pi i/D)$ be the D-th root of unity. For a vertex $u \in V(G)$ with neighbours $v_1 < \cdots < v_D$, define for $j \in [D-1]_0$:*

$$|\widehat{\psi^j}(u)\rangle := \frac{1}{\sqrt{D}} \sum_{i=0}^{D-1} \omega_D^{i\cdot j}|u, v_{i+1}\rangle.$$

Then these vectors form an orthonormal set and they can be used to define the alternative Fourier neighbourhood of dimension D *of the vertex u:*

$$\widehat{\Psi}_\star^D(u) = \{|\widehat{\psi}_u^1\rangle, \dots, |\widehat{\psi}_u^{D-1}\rangle\}.$$

For any $u \in V_{\text{odd}}\backslash\{t\}$, we let the set of alternative neighbourhoods be the alternative Fourier neighbourhood $\Psi_\star(u) = \widehat{\Psi}_\star^D(u)$. It is straightforward to verify that $|\psi_\star(u)\rangle \in \text{span}\{\widehat{\Psi}_\star^D(u)\}$ for any $u \in V_{\text{odd}}\backslash\{t\}$, due to (6.25) that

$$\langle\widehat{\psi}_u^0|\psi_\star(u)\rangle \propto -\ell \cdot e_{k+1} + (D-\ell)e_k = D \cdot e_k - \ell(e_k + e_{k+1}) = D(e_k - \ell \cdot s_k) = 0.$$

This means that for every $u \in V_{\text{odd}}\backslash\{t\}$, generating $\Psi_\star(u)$ does not depend on u (apart from making the query to obtain its neighbours) and it will allow us to efficiently implement the quantum walk operator $U_{\mathcal{A}^{\text{alt}}\mathcal{B}}$, which we prove in Lemma 6.3.8.

Before we do this however, we first finish showing that we can apply Theorem 6.2.10 to this electrical network. On these one-dimensional random hierarchical graphs, it turns out that the s-t electrical flow (see (6.29)) actually matches the alternative s-t electrical flow with respect to our alternative neighbourhoods. We verify this via the following lemma:

Claim 6.3.7. *For any $u \in V(G)$, define $|\theta_u\rangle = (|u\rangle\langle u| \otimes I)|\theta\rangle$. If $u \in V_{\text{even}}$, then $|\theta_u\rangle \propto |\widehat{\psi^0}(u)\rangle$. If $u \in V_{\text{odd}}$, then $|\theta_u\rangle \propto \sum_{v\in\Gamma(u)} \theta_{u,v}|u, v\rangle$. As a consequence, for every $u \in V$ and $|\psi_\star\rangle \in \Psi_\star(u)$ the state $|\theta_u\rangle$ satisfies $\langle\psi_\star|\theta_u\rangle = 0$.*

Proof. By (3.29) we obtain that

$$\begin{aligned} |\theta\rangle &= \frac{1}{\sqrt{2\mathcal{R}_{s,t}}} \sum_{(u,v)\in\vec{E}(G)} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}(|u, v\rangle + |v, u\rangle) \\ &= \frac{1}{\sqrt{2\mathcal{R}_{s,t}}} \sum_{k=1}^{n} \sum_{(u,v)\in\vec{E}_k} (-1)^{\Delta_{u,v}} \frac{1}{e_k\sqrt{\mathsf{w}_k}}(|u, v\rangle + |v, u\rangle), \end{aligned} \tag{6.32}$$

meaning that for any $u \in V(G)$ the state $|\theta_u\rangle$ is equal to

$$|\theta_u\rangle = \frac{1}{\sqrt{2\mathcal{R}_{s,t}}} \sum_{v\in\Gamma(u)} \frac{\theta_{u,v}}{\sqrt{\mathsf{w}_{u,v}}}|u, v\rangle.$$

Let k such that $u \in S_k$ and let $v_1, \dots, v_\ell \in \Gamma(u) \cap S_{k-1}$ be the neighbours of u that lay in the node S_{k-1} and similarly let $v_{\ell+1}, \dots, v_D \in \Gamma(u) \cap S_{k+1}$ be the neighbours of u that lay in the node S_{k+1}, where $\ell = \frac{D}{1+r_k}$, which is in fact an integer. This means by (6.29) that $\theta_{u,v_i} = (-1)^{\Delta_{u,v_i}}/e_k$ for $i \in [\ell]$ and $\theta_{u,v_i} = (-1)^{\Delta_{u,v_i}}/e_{k+1}$ for $i \in [D]\backslash[\ell]$. If $u \in V_{\text{even}}$, then the weights (see (6.26)) satisfy

$$\sqrt{\frac{\mathsf{w}_{k+1}}{\mathsf{w}_k}} = \frac{e_k}{e_{k+1}} = \frac{1}{r_k},$$

meaning

$$\frac{1}{e_k\sqrt{\mathsf{w}_k}} = \frac{1}{e_{k+1}\sqrt{\mathsf{w}_{k+1}}}.$$

Additionally, since $u \in V_{\text{even}}$, it holds that $(-1)^{\Delta_{u,v_i}}(-1)^{\Delta_{u,v_j}} = -1$ for any $i \in [\ell]$ and $j \in [D]\backslash[\ell]$, meaning $|\theta_u\rangle \propto |\widehat{\psi^0}(u)\rangle$. Since for $u \in V_{\text{even}}$ we defined $\Psi_\star(u)$ to be the alternative Fourier neighbourhood (see Definition 6.3.6) and the Fourier basis states form an orthonormal basis, it follows that $\langle\psi_\star|\theta_u\rangle \propto \langle\psi_\star|\widehat{\psi^0}(u)\rangle = 0$.

Now if instead $u \in V_{\text{odd}}$, then we know that $\mathsf{w}_k = \mathsf{w}_{k+1}$ and $(-1)^{\Delta_{u,v_i}}(-1)^{\Delta_{u,v_j}} = 1$ for any $i \in [\ell]$ and $j \in [D]\backslash[\ell]$. So $|\theta_u\rangle \propto \sum_{v\in\Gamma(u)} \theta_{u,v}|u,v\rangle$. Since for $u \in V_{\text{odd}}$ we defined $\Psi_\star(u) = \{|\psi_\star(u)\rangle = |\widehat{\psi^0}(u)\rangle\}$, it follows by the conservation of the flow θ that $\langle\psi_\star(u)|\theta_u\rangle = \sum_{v\in\Gamma(u)} \theta_{u,v} = 0$. □

We can therefore apply Theorem 6.2.10, but to analyse the resulting complexity we still need to compute the cost of implementing the quantum walk operator. We do this via the following lemma, which is essentially similar to Lemma 4.4.5, but it tackles the more general case where every vertex (except s and t) is of degree D:

Lemma 6.3.8. *The quantum walk operator $U_{\mathcal{A}^{\text{alt}}\mathcal{B}}$ as defined in (4.1) can be implemented in $O(1)$ queries to O_G and $O(nD)$ elementary operations.*

Proof. Let

$$\mathcal{H}' = \text{span}\{|j\rangle|u,v\rangle : j \in [D-1]_0, u \in V(G), v \in \Gamma(u) \cup \{\perp\}\},$$

so in particular $|0\rangle \otimes \mathcal{H} \subset \mathcal{H}'$. Here $\mathcal{H}$ is as in (6.1), meaning we once again consider the modified graph G' where s_0 is connected to s with an edge of weight $\sqrt{w_0}$ and we have to reflect around the star state of s with respect to G'.

We first describe how to implement $2\Pi_{\mathcal{A}} - I$. We describe a unitary $U_\star$ on $\mathcal{H}'$, and in particular, its behaviour on states of the form $|j\rangle|u,\perp\rangle$, where $j = 0$ whenever $u \in V_{\text{even}} \cup \{t\}$, and $j \in [D-1]$ whenever $u \in V_{\text{odd}} \setminus \{t\}$. We begin by querying the neighbours of u in an auxiliary register, Q, initialised to $|0\rangle$ using O_G:

$$|j\rangle|u,\perp\rangle|0\rangle_Q \mapsto |j\rangle|u,\perp\rangle|v_1,\dots,v_D\rangle_Q$$

where if $u \in \{s,t\}$, $v_1 < \cdots < v_{D-1}$ and $v_D = \perp$ (which we can interpret as the extra vertex s_0 when $u = s$), and otherwise, since we assume $u \in V(G)$, $v_1 < \cdots < v_D$ are the neighbours of u. We initialise an auxiliary register A, and compute a trit $|a\rangle_A$ for $a \in \{0,1,2\}$ as follows, to determine what happens next. If $v_D \neq \perp$, then $a = 0$. Else if $u = s$, we let $a = 1$. Else if $v_3 = \perp$ but $u \neq s$, so $u = t$, we let $a = 2$. Controlled on $|0\rangle_A$, apply QFT_D to $|j\rangle$ to get

$$|\widehat{j}\rangle = \frac{1}{\sqrt{D}}(|1\rangle + \omega_D^j|2\rangle + \cdots + \omega_D^{(D-1)j}|D\rangle),$$

which requires $O(\log^2(D))$ elementary operations (see Definition 2.2.1). Then, still conditioned on $|0\rangle_A$, swap the first and third registers, so now the first register contains $|\perp\rangle$, and then perform $|\perp\rangle \mapsto |0\rangle$ on the first register to obtain

$$|0\rangle|u\rangle|\widehat{j}\rangle|0\rangle|v_1,\ldots,v_D\rangle_Q|0\rangle_A.$$

Then, conditioned on the value in the $|\widehat{j}\rangle$ register, we can copy over the first, second or third value in the $|v_1,\ldots,v_D\rangle$ register to get

$$\frac{1}{\sqrt{D}}|0\rangle|u\rangle\left(|1\rangle|v_1\rangle+\omega_D^j|2\rangle|v_2\rangle+\cdots+\omega_D^{(D-1)j}|D\rangle|v_D\rangle\right)|v_1,\ldots,v_D\rangle_Q|0\rangle_A.$$

This requires $O(nD)$ basic operations, since $\ell=\Theta(n)$. We can uncompute the value $|i\rangle$ in $|i\rangle|v_i\rangle$ by referring to the last register to learn v_i's position, and then we are left with:

$$|0\rangle|\widehat{\psi^j}(u)\rangle|v_1,\ldots,v_D\rangle_Q|0\rangle_A.$$

Next, we control on $|1\rangle_A$, meaning $u=s$. In that case, we assume that $j=0$. Using $v_1,\ldots,v_{D_1}$ in the last register, we can map $|\perp\rangle$ to a state proportional to

$$\sqrt{\mathsf{w}_0}|v_0\rangle+\sqrt{w_1}|v_1\rangle+\cdots+\sqrt{w_1}|v_{D-1}\rangle$$

to get

$$|0\rangle|\psi_\star^{G'}(s)\rangle|v_1,\cdots,v_D\rangle_Q|1\rangle_A.$$

Lastly, we control on $|2\rangle_A$, meaning $u=t$. Using $v_1,\ldots,V_{D-1}$, we map $|\perp\rangle$ to a state proportional to:

$$|v_1\rangle+\cdots+|v_{D-1}\rangle$$

to get

$$|0\rangle|\psi_\star^{G'}(t)\rangle|v_1,\cdots,v_D\rangle_Q|2\rangle_A.$$

We can uncompute the register A, since the registers containing u, and $v_1,\ldots,v_D$ haven't changed. Since the register containing u has not changed, we can uncompute the register $|v_1,\cdots,v_D\rangle_Q$ using another call to O_G. Then we have performed a map, $U_\star$ that acts, for $j=0$ when $u\in V_{\text{even}}\cup\{t\}$ and $j\in[D-1]$ when $u\in V_{\text{odd}}\setminus\{t\}$, as $|j\rangle|u,\perp\rangle\mapsto|0\rangle|\widehat{\psi^j}(u)\rangle$, where, using (6.30), for all $u\in V_{\text{even}}\cup\{t\}$,

$$\operatorname{span}\{|\widehat{\psi^0}(u)\rangle\}=\operatorname{span}\{|\psi_\star^{G'}(u)\rangle\}=\operatorname{span}\{\Psi_\star(u)\}$$

and using Definition 6.3.6 for all $u\in V_{\text{odd}}\setminus\{t\}$:

$$\operatorname{span}\{|\widehat{\psi^1}(u)\rangle,\ldots,|\widehat{\psi^{D-1}}(u)\rangle\}=\operatorname{span}\{\widehat{\Psi}_\star^D(u)\}=\operatorname{span}\{\Psi_\star(u)\}.$$

Thus, $U_\star$ maps the subspace

$$\mathcal{L}:=\operatorname{span}\{|0,u,\perp\rangle:u\in(V_{\text{even}}\cup\{s\})\setminus\{t\}\}\cup\{|j,u,\perp\rangle:u\in V_{\text{odd}}\setminus\{t\},j\in[D-1]\}$$

of $\mathcal{H}'$ to $|0\rangle\otimes\operatorname{span}\{\Psi^{\mathcal{A}}\}\cong\mathcal{A}$, and thus, $2\Pi_{\mathcal{A}}-I=U_\star\,(2\Pi_{\mathcal{L}}-I)\,U_\star^\dagger$.

We describe how to implement $2\Pi_{\mathcal{L}}-I$. First, initialise four auxiliary flag qubits $|0\rangle_{F_1}|0\rangle_{F_2}|0\rangle_{F_3}|0\rangle_{F_4}$. For a computational basis state $|j\rangle|u,v\rangle$ of $\mathcal{H}'$, by assumption (which is removed at the end of this section) we can efficiently check whether u is in V_{odd} or V_{even}, and we can check whether $u=s$ or $u=tM$ in $O(n)$ cost. If $u\in V_{\text{even}}\cup\{s\}\setminus\{t\}$, we check if the first register is 0, and if not, flip F_1 to get $|1\rangle_{F_1}$. If $u\in V_{\text{odd}}\setminus\{t\}$, we check if the first register is 1 or 2, and if not, flip F_2 to get $|1\rangle_{F_2}$. If the last register is not $\perp$, flip F_3 to get $|1\rangle_{F_3}$. Lastly if $u\in\{t\}\cup\{s_0\}$, flip F_4 to get $|1\rangle_{F_4}$. Reflect if any flag is set, and then uncompute all flags. This can all be done in $O(n)$ basic operations.

Next, we describe how to implement $2\Pi_{\mathcal{B}} - I$. We describe a unitary $U_{\mathcal{S}}$ on $\mathcal{H}'$, and in particular, its behaviour on states of the form $|1\rangle|u,v\rangle$ for $(u,v) \in E(G')$ with $u < v$. First, apply a Hadamard gate to the first register, and then, controlled on its value, swap the second two registers to get

$$(|0\rangle|u,v\rangle - |1\rangle|v,u\rangle)/\sqrt{2}.$$

We can uncompute the first register by adding in a bit indicating if the last two registers are in sorted order, to get

$$|0\rangle\frac{1}{\sqrt{2}}(|u,v\rangle - |v,u\rangle) \in \begin{cases} \mathrm{span}\{|0\rangle|\overline{\psi}_{u,v}\rangle\} & \text{if } (u,v) \in \overrightarrow{E}(G'), \\ \mathrm{span}\{|0\rangle|\overline{\psi}_{v,u}\rangle\} & \text{if } (v,u) \in \overrightarrow{E}(G'). \end{cases}$$

Thus $U_{\mathcal{S}}$ maps

$$\mathcal{L}' := \mathrm{span}\{|1\rangle|u,v\rangle : (u,v) \in E(G'), u < v\}$$

to $\mathrm{span}\{|0\rangle|\overline{\psi}_{u,v}\rangle : (u,v) \in \overrightarrow{E}(G')\} \cong \mathcal{B}$, and so $2\Pi_{\mathcal{B}} - I = U_{\mathcal{S}}(2\Pi_{\mathcal{L}'} - I)U_{\mathcal{S}}^{\dagger}$. To implement $(2\Pi_{\mathcal{L}'} - I)$, it is enough to check that the first register is 1, and u and v are in sorted order (we know $(u,v) \in E(G')$ by the structure of $\mathcal{H}'$). This can be done in $O(n)$ basic operations. □

6.3.3 The algorithm

In this section, we provide a quantum algorithm that approximates the s-t electrical flow state and samples from it to find the ending vertex $t \in V$ in a one-dimensional random hierarchical graph, to solve Problem 6.3.5. As an example of such a one-dimensional random hierarchical graph, we then apply our algorithm to the welded trees graph from Section 4.4 to solve Problem 4.4.1.

Algorithm 1 Solving the one-dimensional random hierarchical graph problem

Require: One-dimensional random hierarchical graph $G = (V, E)$ with adjacency list oracle O_G, the ℓ-bit string corresponding to the starting vertex $s \in V$, a success probability parameter δ.

Ensure: The ℓ-bit string corresponding to the ending vertex $t \in V$.

1. Set $i = 1$ and $T_1 = \Theta(\mathcal{R}_{s,t}D\mathsf{w}_n)$.
2. Run phase estimation on the quantum walk operator $U_{\mathcal{A}^{\mathsf{alt}}\mathcal{B}}$ and state $|\psi_0\rangle = \frac{1}{\sqrt{2}}(|s_0,s\rangle + |s,s_0\rangle)$ to precision $O(\frac{\epsilon^2}{\sqrt{\mathsf{ET}}})$, where $\epsilon = \frac{1}{2\mathcal{R}_{s,t}D\mathsf{w}_n}$, and measure the phase register. If the output is "0", return the resulting state $|\widetilde{\theta}\rangle$.
3. Measure $|\widetilde{\theta}\rangle$ to obtain an outcome $|u,v\rangle$, representing the edge $(u,v) \in E$. Check if u or v is equal to t and if this is the case, return the ℓ-bit string corresponding to t. Otherwise, if $i < T_1$, increment i by 1 and return to Step 2.

Theorem 6.3.9. *Let G be a D-regular one-dimensional random hierarchical graph on the line supergraph of length n with edge ratios $r_0, \ldots, r_{n-1}$. Let $\mathsf{w}_n = \prod_{k=1}^{\lfloor n/2 \rfloor}(\frac{1}{r_{2k-1}})^2$ and let each vertex in G be identified by an ℓ-bit string where $\ell = \Theta(n)$. Given access to an adjacency list oracle O_G to the graph G, there exists a quantum algorithm that solves Problem 6.3.5, i.e. recovering the ℓ-bit string associated with the root t, with bounded error using*

$$O(\sqrt{\mathsf{ET}_s}\mathcal{R}_{s,t}^3 D^3 \mathsf{w}_n^3) \text{ queries}, \qquad O(n\sqrt{\mathsf{ET}_s}\mathcal{R}_{s,t}^3 D^4 \mathsf{w}_n^3) \text{ time}.$$

Proof. The proof consists of a cost and success probability analysis of Algorithm 1. By Theorem 6.2.10, the phase estimation algorithm in Step 2 succeeds and returns an approximation of $|\theta\rangle$ with bounded error.

Suppose that we had a perfect copy of $|\theta\rangle$, then after measuring it we would obtain an edge $(u, v) \in E$ containing the vertex t with probability

$$\frac{1}{\mathcal{R}_{s,t}} \sum_{u \in \Gamma(t)} \frac{\theta_{u,t}^2}{\mathsf{w}_{u,t}} = \frac{1}{\mathcal{R}_{s,t} D \mathsf{w}_n}.$$

Instead, we have access to a state $|\widetilde{\theta}\rangle$, which by Theorem 6.2.10 satisfies

$$\frac{1}{2} \left\| |\widetilde{\theta}\rangle\langle\widetilde{\theta}| - |\theta\rangle\langle\theta| \right\|_1 \leq \epsilon = \frac{1}{2\mathcal{R}_{s,t} D \mathsf{w}_n}.$$

Hence, by measuring $|\widetilde{\theta}\rangle$, we obtain an edge $(u, v) \in E$ that contains the vertex t with probability at least $\Theta\left(\frac{1}{\mathcal{R}_{s,t} D \mathsf{w}_n}\right)$. The probability that at least one out of the at most $T_1 = \Theta(\mathcal{R}_{s,t} D \mathsf{w}_n)$ repetitions succeeds in returning the vertex t is therefore constant.
For the cost of Step 2, we require

$$O\left(\frac{\sqrt{\mathsf{ET}_s^{\mathsf{alt}}}}{\epsilon^2}\right) = O(\sqrt{\mathsf{ET}_s} \mathcal{R}_{s,t}^2 D^2 \mathsf{w}_n^2)$$

calls to $U_{\mathcal{A}^{\mathsf{alt}}\mathcal{B}}$, since the s-t electrical flow matches the alternative s-t electrical flow, meaning $\mathsf{ET}_s^{\mathsf{alt}} = \mathsf{ET}_s$. By Lemma 6.3.8, each such call has a cost of $O(1)$ queries and $O(nD)$ elementary operations. Since we can set up the initial state $|\psi_0\rangle$ in $O(\ell) = O(n)$ elementary operations (and no queries) and we run at most $T_1 = \Theta(\mathcal{R}_{s,t} D \mathsf{w}_n)$ iterations of phase estimation, we find that the total contribution of Step 2 to the cost is

$$O(\sqrt{\mathsf{ET}_s} \mathcal{R}_{s,t}^3 D^3 \mathsf{w}_n^3) \text{ queries}, \qquad O(n\sqrt{\mathsf{ET}_s} \mathcal{R}_{s,t}^3 D^4 \mathsf{w}_n^3) \text{ time}.$$

For the cost of Step 3, we must only verify whether u or v is equal to t, which can be done in zero queries and $O(\ell) = O(n)$ elementary operations. So the cost of Step 2 dominates the total cost of the algorithm. □

6.3.4 Welded trees

As an example of a one-dimensional hierarchical graph, we apply Theorem 6.3.9 to the welded trees. This shows how sampling from the electrical flow can recover the exponential speedup from Theorem 4.4.2. Our parameterisation will be slightly different compared to Section 4.4 since we view the welded trees as a one-dimensional hierarchical graph of length n. This means that the graph consists of two full binary trees of depth h and contains $2^{h+2} - 2$ vertices. The leaves of both trees are connected via two disjoint perfect matchings, resulting in a one-dimensional random hierarchical graph on the line supergraph of length $n = 2h + 1$. For each $k \in [2h + 1]_0$, every node S_k contains

$$s_k = \begin{cases} 2^k & \text{if } k \in [h]_0 \\ 2^{2h+1-k} & \text{if } k \in \{h+1, \ldots, 2h+1\}, \end{cases}$$

vertices, meaning that its edge ratios are equal to

$$r_k = \begin{cases} 2 & \text{if } k \in [h] \\ \frac{1}{2} & \text{if } k \in \{h+1, \ldots, 2h+1\}. \end{cases}$$

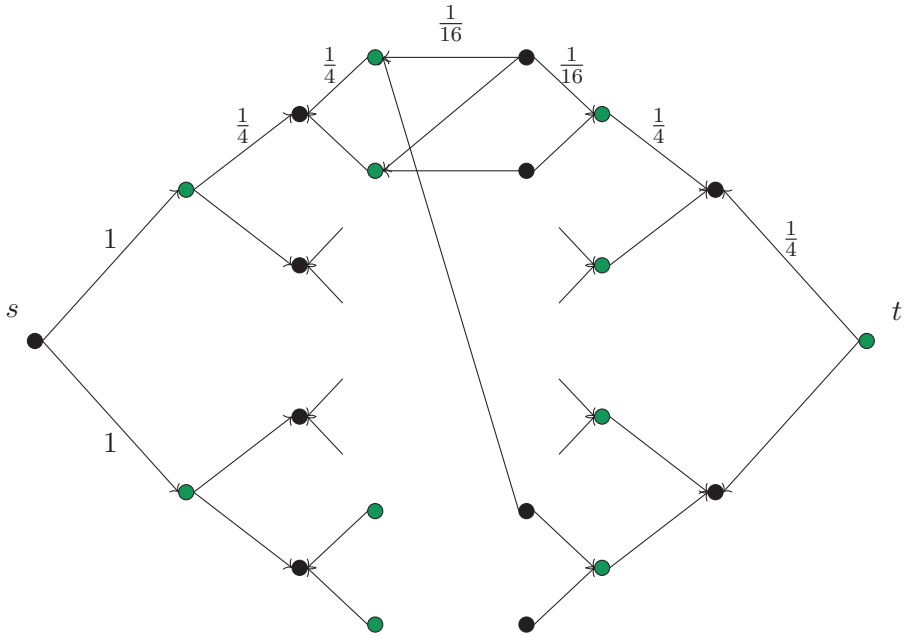


Figure 6.5: The welded trees graph with depth $h = 3$: the black vertices are the vertices in V_{even}, where the edge directions are reversed and where all adjacent edges have the same weight and direction.

Since $V = 2^{h+2} - 2$, we find that $\ell = 2h$ satisfies $2^\ell \gg |V|$, meaning each vertex is assigned a $2h$-bit string as an identifier. Before we apply Theorem 6.3.9 to the welded trees graph, we first obtain a little more insight about its weights w_k. The weight assignment and edge directions are slightly different from Section 4.4, as we will apply the ones from (6.26) and (6.27) and we assume without loss of generality that h is odd:

$$\mathsf{w}_k = \begin{cases} 2^{-2\lfloor k/2 \rfloor} & \text{if } k \in [h+1] \\ 2^{-2(h+1-\lfloor k/2 \rfloor)} & \text{if } k \in \{h+2, \dots, 2h+1\}. \end{cases} \tag{6.33}$$

This is visualised in Figure 6.5.

Corollary 6.3.10. *Given an adjacency list oracle O_G to the welded trees graph G, there exists a quantum algorithm that solves Problem 4.4.1 with bounded error and cost*

$$O\left(n^3\right) \text{ queries}, \qquad O\left(n^4\right) \text{ time}.$$

Proof. The proof follows by bounding the quantities $\mathcal{R}_{s,t}, D, \mathsf{w}_n$ and ET in Theorem 6.3.9. From (6.33), we see that $\mathsf{w}_n = 1/4$. Additionally, the effective resistance from (6.28) can be computed to find that $\mathcal{R}_{s,t} = \Theta(n)$. Since $D = 3$ and $\mathsf{p}_s = \mathcal{R}_{s,t}$ is the largest potential value, we only need to bound ET using (3.42):

$$\mathsf{ET}_s = \frac{1}{\mathcal{R}_{s,t}} \sum_{k=0}^{n} \sum_{u \in S_k} \mathsf{p}_u^2 \mathsf{w}_u = O(n^2). \qquad \square$$

The result of Corollary 6.3.10 is worse than our previous result from Theorem 4.4.2, since by approximating the electrical flow, we infer much more information than is actually needed to recover the bit string associated with t, but it exemplifies how sampling from the electrical flow can provide an exponential speedup.

6.4 An exponential speedup for pathfinding

In this section, we show that the electrical flow in a multidimensional electrical network can also be used to demonstrate an exponential quantum-classical separation for the pathfinding problem relative to an oracle. We achieve this by constructing and sampling from the s-t *alternative* electrical flow that we defined in Definition 6.2.2, which is the flow achieving minimal energy out of all unit s-t flows satisfying Alternative Kirchhoff's Law. We show that this flow also satisfies Alternative Ohm's Law by explicitly constructing the alternative potential $\mathsf{p}^{\mathsf{alt}}$. Throughout this section, we assume that the parameter n is odd for readability, though the results can be slightly modified to also hold for even n.

We construct a family of regular graphs, which we name *welded trees circuit graphs* (see Figure 6.6), that have exponentially many vertices in n. Our framework allows us to approximate the alternative s-t electrical flow state $|\theta^{\mathsf{alt}}\rangle$ in polynomial time for these types of graphs. We then show that the overlap between each edge of an explicit s-t path and the s-t alternative electrical flow is at least inverse polynomial. Therefore, by sampling from a polynomial number of copies of $|\theta^{\mathsf{alt}}\rangle$, we can obtain a polynomial-sized subgraph that contains an s-t path. A classical algorithm such as Breadth First Search can then be used to traverse the subgraph and output this s-t path.

We explicitly compute what the s-t alternative electrical flow looks like in the welded trees circuit graph. Unlike the s-t alternative electrical flow generated in one-dimensional unweighted random hierarchical networks, which matches the actual electrical flow in the electrical network, the s-t alternative electrical flow in our pathfinding example is significantly different from any "real" electrical flow.We shall see that the s-t alternative electrical flow in the welded trees circuit graph essentially follows the s-t alternative electrical flow in the simple example graph G_1, visualised in Figure 6.7. The additional alternative neighbourhoods ensure that the s-t alternative electrical flow splits evenly at the coloured vertices, which employ alternative Fourier neighbourhoods. This is a direct consequence of Alternative Kirchhoff's Law. Although multiple such s-t flows exist, the s-t alternative electrical flow is the one that minimises the energy among all possible flows.

6.4.1 Example graph G_1

Since the welded trees circuit graph that we will try to find an s-t path for is quite large, we start by analysing the s-t alternative flow and alternative potential for smaller graphs that will form the building blocks for the larger graph. We begin with a network $G_1 = (V, E, \mathsf{w})$, whose vertex set is given by $V = \{s, v_1, \ldots, v_8, t\}$. We have visualised G_1, with its directed edge set and weights in Figure 6.7. These directions and weights give rise to the star states $|\psi_\star(u)\rangle$ for each $u \in V$, but we will also consider the following additional alternative neighbourhoods for the vertices $v_2, v_3, v_8 \in V$:

$$
\begin{aligned}
|\overline{\psi}^{\mathsf{alt}}_{v_2}\rangle &= \sqrt{\frac{2}{3}}\left(-|v_2, v_4\rangle + \frac{1}{2}|v_2, s\rangle + \frac{1}{2}|v_2, v_5\rangle\right), \\
|\overline{\psi}^{\mathsf{alt}}_{v_3}\rangle &= \sqrt{\frac{2}{3}}\left(\frac{1}{2}|v_3, v_1\rangle - |v_3, v_6\rangle + \frac{1}{2}|v_3, v_7\rangle\right), \\
|\overline{\psi}^{\mathsf{alt}}_{v_8}\rangle &= \sqrt{\frac{2}{3}}\left(\frac{1}{2}|v_8, t\rangle - |v_8, v_5\rangle + \frac{1}{2}|v_8, v_6\rangle\right).
\end{aligned}
\tag{6.34}
$$

These alternative neighbourhoods are the same as those used for the welded trees graph in Section 4.4, ensuring that for $u \in \{v_2, v_3, v_8\}$, we have $\Psi_\star(u) = \widehat{\Psi}^3_\star(u)$, i.e. the alternative Fourier neighbourhood of dimension 3 (see Definition 6.3.6).

Any s-t alternative unit flow θ^{alt} must be conserved at every vertex and satisfy $\theta^{\mathsf{alt}}_{s,v_1} = x$, $\theta^{\mathsf{alt}}_{s,v_2} = y$ for some $x, y \in [0, 1]$ such that $x + y = 1$. For θ^{alt} to also satisfy Alternative

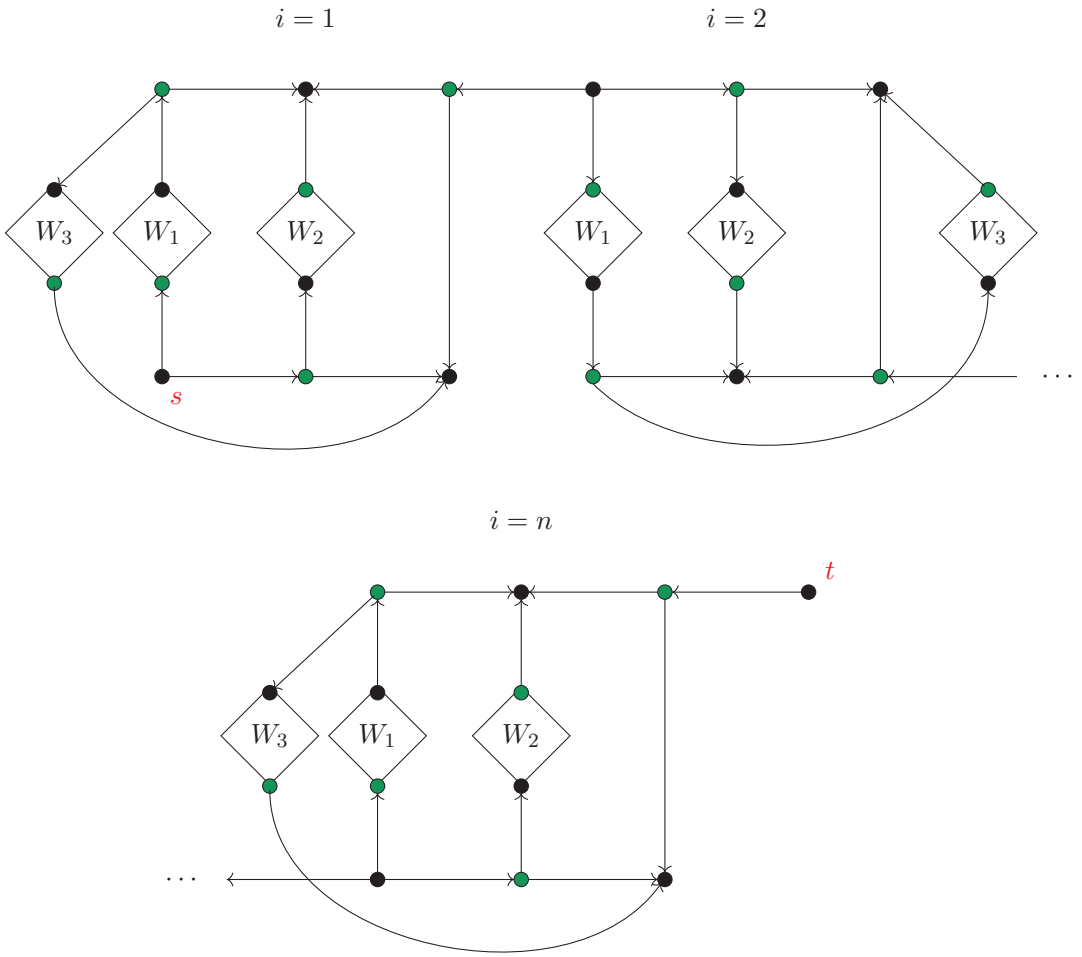


Figure 6.6: The welded trees circuit graph. Each W_1, W_2, W_3 represents a welded trees graph of depth n.

Kirchhoff's Law (see Definition 6.2.1), the flow coming into any vertex v_2, v_3, v_8 through the edge with the highest weight must be evenly distributed along the other two neighbours. This is visualised in Figure 6.7, and we end up with a single parameter x (because $y = 1 - x$) that parametrises all possible s-t alternative unit flows θ^{alt} on G_1. The energy of each such θ^{alt} can be explicitly calculated as $\mathcal{E}(\theta^{\mathsf{alt}}) = 5y^2 + 4x^2 + 3$, and the energy is minimised for $x = 5/9$, resulting in the alternative effective resistance $\mathcal{R}^{\mathsf{alt}}_{s,t} = 47/9$.

We now explicitly construct the alternative potential $\mathsf{p}^{\mathsf{alt}}$ corresponding to this s-t alternative electrical flow, satisfying $\mathsf{p}^{\mathsf{alt}}_{s,v_1} = \mathsf{p}^{\mathsf{alt}}_{s,v_2} = \mathcal{R}^{\mathsf{alt}}_{s,t} = 47/9$, $\mathsf{p}^{\mathsf{alt}}_{t,v_8} = 0$, and Alternative Ohm's Law (see Definition 6.2.4). We do this by constructing the states $|\mathsf{p}^{\mathsf{alt}}_{|u}\rangle \in \mathrm{span}\{\Psi_\star(u)\}$ from (6.9):

$$|\mathsf{p}^{\mathsf{alt}}_{|s}\rangle = \frac{47}{9}|s, v_1\rangle + \frac{47}{9}|s, v_2\rangle, \qquad |\mathsf{p}^{\mathsf{alt}}_{|v_1}\rangle = -\frac{43}{9}|v_1, s\rangle + \frac{43}{9}|v_1, v_3\rangle,$$
$$|\mathsf{p}^{\mathsf{alt}}_{|v_2}\rangle = -\frac{42}{9}|v_2, s\rangle + \frac{19}{9}|v_2, v_4\rangle + \frac{23}{9}|v_2, v_5\rangle, \qquad |\mathsf{p}^{\mathsf{alt}}_{|v_3}\rangle = -\frac{39}{9}|v_3, v_1\rangle + \frac{26}{9}|v_3, v_7\rangle + \frac{13}{9}|v_3, v_6\rangle,$$
$$|\mathsf{p}^{\mathsf{alt}}_{|v_4}\rangle = -|v, u\rangle + |v, t\rangle, \qquad |\mathsf{p}^{\mathsf{alt}}_{|v_5}\rangle = -2|v_5, v_8\rangle - 2|v_5, v_7\rangle - 2|v_5, v_2\rangle,$$
$$|\mathsf{p}^{\mathsf{alt}}_{|v_6}\rangle = -|v_6, v_8\rangle - |v_6, v_4\rangle - |v_6, v_3\rangle, \qquad |\mathsf{p}^{\mathsf{alt}}_{|v_7}\rangle = -\frac{22}{9}|v_7, v_3\rangle + \frac{22}{9}|v_7, v_5\rangle,$$
$$|\mathsf{p}^{\mathsf{alt}}_{|v_8}\rangle = -|v_8, t\rangle + 0|v_8, v_6\rangle) + |v_8, v_5\rangle, \qquad |\mathsf{p}^{\mathsf{alt}}_{|t}\rangle = 0|t, v_8\rangle.$$

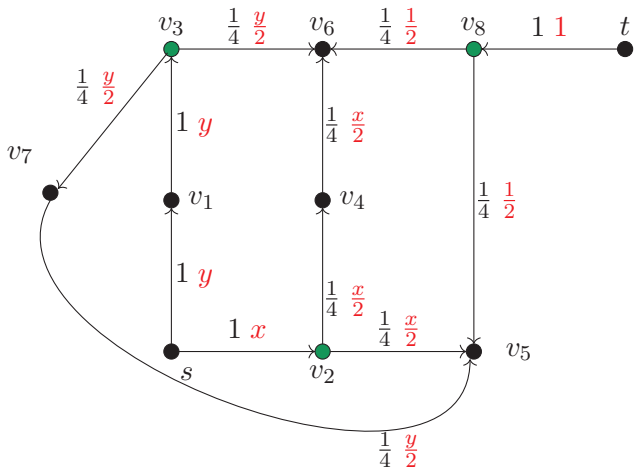


Figure 6.7: The graph G_1 with corresponding edge directions where the coloured vertices have an additional alternative neighbour as defined in (6.34). For each directed edge (u, v), the weights $\mathsf{w}_{u,v}$ are denoted in black and the flow values $\theta^{\mathsf{alt}}_{u,v}$ in red for any valid unit s-t alternative flow are parametrised by x and $y = 1 - x$.

It is straightforward to verify that these states indeed satisfy Alternative Ohm's Law as well as the equations $\mathsf{p}^{\mathsf{alt}}_{s,v_1} = \mathsf{p}^{\mathsf{alt}}_{s,v_2} = 47/9$, $\mathsf{p}^{\mathsf{alt}}_{t,v_8} = 0$. It is also clear that $|\mathsf{p}^{\mathsf{alt}}_{|u}\rangle \in \mathrm{span}\{\Psi_\star(u)\}$ for every u without additional alternative neighbourhoods, i.e. for $u \in \{s, v_1, v_4, v_5, v_6, v_7, t\}$, since all outgoing edge potentials are the same. For $u \in \{v_2, v_3, v_8\}$, we can confirm that $|\mathsf{p}^{\mathsf{alt}}_{|u}\rangle \in \mathrm{span}\{\Psi_\star(u)\}$ by calculating that all the amplitudes of $|\mathsf{p}^{\mathsf{alt}}_{|u}\rangle$ sum to 0.

6.4.2 Example graph G_2

The second example graph, $G_2 = (V, E, \mathsf{w})$, depicted in Figure 6.8, is constructed by combining the graph G_1 (see Figure 6.7) with three welded trees graphs, W_1, W_2, W_3 (see Figure 6.5). The "starting" roots of these three welded trees graphs are w_1, w_4, and w_6, respectively. In the following section, we will use multiple instances of G_2 to assemble the final graph for our pathfinding example.

As seen in Section 6.3.4, the welded trees graph is an example of a one-dimensional random hierarchical graph with nodes $\{S_0, \ldots, S_n\}$. We previously observed that the weight assignments, edge directions, and alternative neighbourhoods in Section 6.3.4 resulted in an s-t electrical flow that matched the s-t alternative electrical flow, as it also satisfied Alternative Kirchhoff's Law. This means that from the perspective of electrical networks, each welded trees graph W_i can be considered as an edge with resistance $\mathcal{R}_i$. We will formalise this intuition shortly. The weights and directions of W_1 in G_2 follow those from Section 6.3.4, so $\mathcal{R}_1 = \mathcal{R}$, where $\mathcal{R}$ is the effective resistance of a welded trees graph of depth n (see (6.28)). For W_2 and W_3, the weights have been scaled down by a factor of $1/4$, and their edge directions are reversed (due to their respective roots being w_4 and w_6), resulting in $\mathcal{R}_2 = \mathcal{R}_3 = 4\mathcal{R}$.

In G_2, the motivation behind the alternative neighbourhoods, edge directions, and weight assignments used in G_1 becomes evident. Similar to the one-dimensional random hierarchical graphs in Section 6.3, these assignments induce a partition of V into two sets: V_{even} and V_{odd}, as indicated by the coloured vertices in Figure 6.8. For each vertex $u \in V_{\mathrm{even}}$, all adjacent edges have uniform weight and direction, which makes it straightforward to generate the star state $|\psi_\star(u)\rangle$. For vertices $u \in V_{\mathrm{odd}} \setminus \{s, t\}$, we have

$|\psi_\star(u)\rangle \in \Psi_\star(u) = \widehat{\Psi}^3_\star(u)$. As in Section 6.3, we can assume without loss of generality that we know whether any given $u \in V$ belongs to V_{even} or V_{odd}, by keeping track of the parity of the distance from s, which starts at 0 and toggles each time the algorithm takes a step.

Since each welded trees graph effectively routes all incoming flow from one root to the other, any s-t alternative unit flow on G_2 behaves like an s-t alternative unit flow on G_1, with the addition that some flow passes through the welded trees graphs. From the configurations in Figure 6.7 and Figure 6.8, it is clear that the energy of an s-t alternative unit flow θ^{alt} can be decomposed into the energy within G_1 plus the energy associated with these welded trees graphs. Therefore, the total energy is given by

$$\mathcal{E}(\theta^{\mathsf{alt}}) = 5y^2 + 4x^2 + 3 + \mathcal{R}_1 y^2 + \mathcal{R}_2 \left(\frac{x}{2}\right)^2 + \mathcal{R}_3 \left(\frac{y}{2}\right)^2 = (2 + 5\mathcal{R})y^2 + (4 + \mathcal{R})x^2 + 3.$$

The energy is minimised by setting $x = \frac{2+5\mathcal{R}}{6+6\mathcal{R}}$, which implies $y = 1 - x = \frac{4+\mathcal{R}}{6+6\mathcal{R}}$. For readability, we keep x in the expression for the alternative effective resistance, but we simplify it using the fact that for these values of x and y, we have $(2 + 5\mathcal{R})y = (4 + \mathcal{R})x$:

$$\mathcal{R}^{\mathsf{alt}}_{s,t} = (2 + 5\mathcal{R})y^2 + (4 + \mathcal{R})x^2 + 3 = (4 + \mathcal{R})(x^2 + xy) + 3 = (4 + \mathcal{R})x + 3.$$

Next, we explicitly construct the alternative potential $\mathsf{p}^{\mathsf{alt}}$ corresponding to this s-t alternative electrical flow, ensuring that $\mathsf{p}^{\mathsf{alt}}_{s,w_1} = \mathsf{p}^{\mathsf{alt}}_{s,v_2} = \mathcal{R}^{\mathsf{alt}}_{s,t} = (4 + \mathcal{R})x + 3$, $\mathsf{p}^{\mathsf{alt}}_{t,v_5} = 0$, and that it satisfies Alternative Ohm's Law. We achieve this by constructing the states $|\mathsf{p}^{\mathsf{alt}}_{|u}\rangle \in \operatorname{span}\{\Psi_\star(u)\}$ from (6.9). For simplicity, we will only show the edges that are visible in Figure 6.8, meaning we will not explicitly write down the amplitudes and basis states for edges inside the welded trees graphs:

$$\begin{aligned}
|\mathsf{p}^{\mathsf{alt}}_{|s}\rangle &= (3 + 5y + 2\mathcal{R}y)|s, w_1\rangle + (3 + 4x + \mathcal{R}x)|s, v_2\rangle, & |\mathsf{p}^{\mathsf{alt}}_{|w_1}\rangle &= -(3 + 4y + 2\mathcal{R}y)|w_1, s\rangle,\\
|\mathsf{p}^{\mathsf{alt}}_{|w_2}\rangle &= (3 + 4y + \mathcal{R}y)|w_2, v_1\rangle, & |\mathsf{p}^{\mathsf{alt}}_{|w_3}\rangle &= -(1 + x + \mathcal{R}x)|w_3, v_2\rangle,\\
|\mathsf{p}^{\mathsf{alt}}_{|w_4}\rangle &= (1 + x)|w_4, v_3\rangle, & |\mathsf{p}^{\mathsf{alt}}_{|w_5}\rangle &= -(4 + 2y + 2\mathcal{R}y)|w_5, v_1\rangle,\\
|\mathsf{p}^{\mathsf{alt}}_{|w_6}\rangle &= (4 + 2y)|w_6, v_4\rangle, & |\mathsf{p}^{\mathsf{alt}}_{|t}\rangle &= 0|t, v_5\rangle,
\end{aligned}$$

$$\begin{aligned}
|\mathsf{p}^{\mathsf{alt}}_{|v_1}\rangle &= -(3 + 3y + \mathcal{R}y)|v_1, w_2\rangle + (2 + 2y + \mathcal{R}y)|v_1, w_5\rangle + (1 + y)|v_1, v_3\rangle,\\
|\mathsf{p}^{\mathsf{alt}}_{|v_2}\rangle &= (1 + 2x + \mathcal{R}x)|v_2, w_3\rangle + (2 + x)|v_2, v_4\rangle - (3 + 3x + \mathcal{R}x)|v_2, s\rangle,\\
|\mathsf{p}^{\mathsf{alt}}_{|v_3}\rangle &= -|v_3, v_5\rangle - |v_3, w_4\rangle - |v_3, v_1\rangle,\\
|\mathsf{p}^{\mathsf{alt}}_{|v_4}\rangle &= -2|v_4, v_5\rangle - 2|v_4, v_2\rangle - 2|v_4, w_6\rangle,\\
|\mathsf{p}^{\mathsf{alt}}_{|v_5}\rangle &= -|v_5, t\rangle + 0|v_5, v_3\rangle) + |v_5, v_4\rangle.
\end{aligned}$$

It is straightforward to verify that these states indeed satisfy Alternative Ohm's Law for all edges outside the welded trees graphs, as well as the conditions $\mathsf{p}^{\mathsf{alt}}_{s,w_1} = \mathsf{p}^{\mathsf{alt}}_{s,v_2} = \mathcal{R}^{\mathsf{alt}}_{s,t}$, given that $(2 + 5\mathcal{R})y = (4 + \mathcal{R})x$ and $\mathsf{p}^{\mathsf{alt}}_{t,v_5} = 0$. It is also evident that $|\mathsf{p}^{\mathsf{alt}}_{|u}\rangle \in \operatorname{span}\{\Psi_\star(u)\}$ for every $u \in \{s, v_3, v_4, t\}$, since all edge potentials are consistent. For $u \in \{v_1, v_2, v_5\}$, we confirm that $|\mathsf{p}^{\mathsf{alt}}_{|u}\rangle \in \operatorname{span}\{\Psi_\star(u)\}$ by calculating that all the amplitudes of $|\mathsf{p}^{\mathsf{alt}}_{|u}\rangle$ sum to 0.

For the edges in the welded trees graphs, we saw in Section 6.3.4 that the s-t alternative electrical flow through each welded trees graph satisfies Ohm's Law. This implies there exist potential values for all vertices (and thus edges), that are lower than the potential at the root where the flow enters, for each welded trees graph that satisfy Alternative Ohm's Law. These potentials are consistent with our potential $\mathsf{p}^{\mathsf{alt}}$ because

$$(\mathsf{p}^{\mathsf{alt}}_{w_1,s} - \mathsf{p}^{\mathsf{alt}}_{w_2,v_1})\frac{1}{y} = (\mathsf{p}^{\mathsf{alt}}_{w_3,v_2} - \mathsf{p}^{\mathsf{alt}}_{w_4,v_3})\frac{1}{x} = \left(\mathsf{p}^{\mathsf{alt}}_{w_5,v_1} - \mathsf{p}^{\mathsf{alt}}_{w_6,v_4}\right)\frac{1}{y} = \mathcal{R}.$$

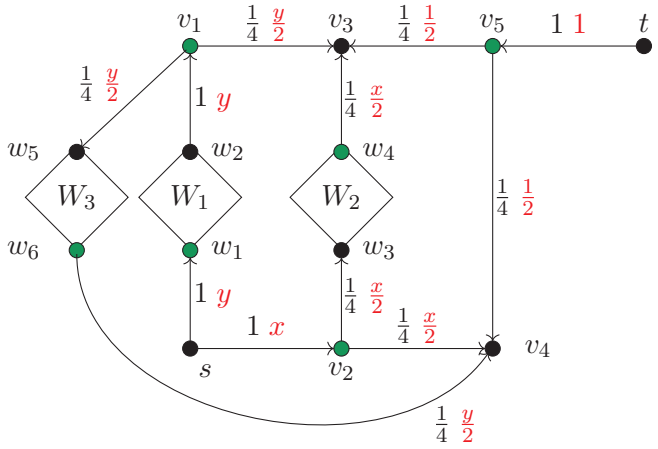


Figure 6.8: The graph G_2 with corresponding edge directions where the coloured vertices are the vertices in V_{odd} and have the alternative neighbourhoods $\Psi_\star(u) = \widehat{\Psi}^3_\star(u)$ (see Definition 6.3.6). Each diamond, indexed by $i \in [3]$ represents a welded trees graph of depth n. For each directed edge (u, v), the weights $\mathsf{w}_{u,v}$ are denoted in black and the flow values $\theta^{\mathsf{alt}}_{u,v}$ in red for any valid unit s-t alternative flow parametrised by x and $y = 1 - x$. The black vertices are the vertices in V_{even}, where the edge directions are swapped and where adjacent edges have the same weight and direction.

Recall from the proof of Corollary 6.3.10 that for a welded trees graph of depth n, we have $\mathcal{R} = \Theta(n)$, which implies that $\mathcal{R}^{\mathsf{alt}}_{s,t} = \Theta(n)$. For the alternative potential, since each edge potential satisfies $\mathsf{p}^{\mathsf{alt}}_{u,v} = O(n)$, we find by (6.9) and (6.19) that:

$$\mathsf{ET}^{\mathsf{alt}}_s = \frac{1}{\mathcal{R}^{\mathsf{alt}}_{s,t}} \sum_{u \in V(G)} \sum_{i=0}^{a_u - 1} \left(\mathsf{p}^{\mathsf{alt}}_{(u,i)}\right)^2 \mathsf{w}_u = \frac{1}{\mathcal{R}^{\mathsf{alt}}_{s,t}} \sum_{(u,v) \in E} (\mathsf{p}^{\mathsf{alt}}_{u,v})^2 \mathsf{w}_{u,v} = O(n^2).$$

We can now invoke Theorem 6.2.10 to approximate the state $|\theta^{\mathsf{alt}}\rangle$. Since the energy along the s-t path $(s, v_2), (v_2, v_4), (v_4, v_5), (v_5, t)$ contains a constant fraction of the total energy $\mathcal{R}^{\mathsf{alt}}_{s,t}$, we could sample from this state to recover an s-t path. However, since this path is of constant length, any classical algorithm can also find this path by performing an exhaustive search of its neighbours in constant time.

6.4.3 The welded trees circuit graph G

The construction of the welded trees circuit graph G involves connecting n graphs that are isomorphic to G_2 (as defined in Section 3.2) sequentially, forming a path structure as shown in Figure 6.10. Each layer in this construction includes three welded trees graphs W_1, W_2, W_3, along with the following set of seven vertices:

$$V_{p,i} := \{v_{p,i,j} : j \in [7]\}.$$

The layers are connected such that for each $i \in [n-1]$, the vertex $v_{p,i,7}$ is identified with $v_{p,(i+1),2}$. The structure of the welded trees graphs is illustrated in Figure 6.9 for the case $j = 1$; for $j \in \{2, 3\}$, the edge directions are reversed. The weights and directions of the edges in these welded trees graphs, as well as in the remaining edges, follow the same pattern described for G_2 in Figure 6.8.

The complete welded trees circuit graph G is shown in Figure 6.10. Due to this construction, all vertices have degree 3, except for the vertices $s = v_{p,1,1}$ and $t = v_{p,n,7}$.

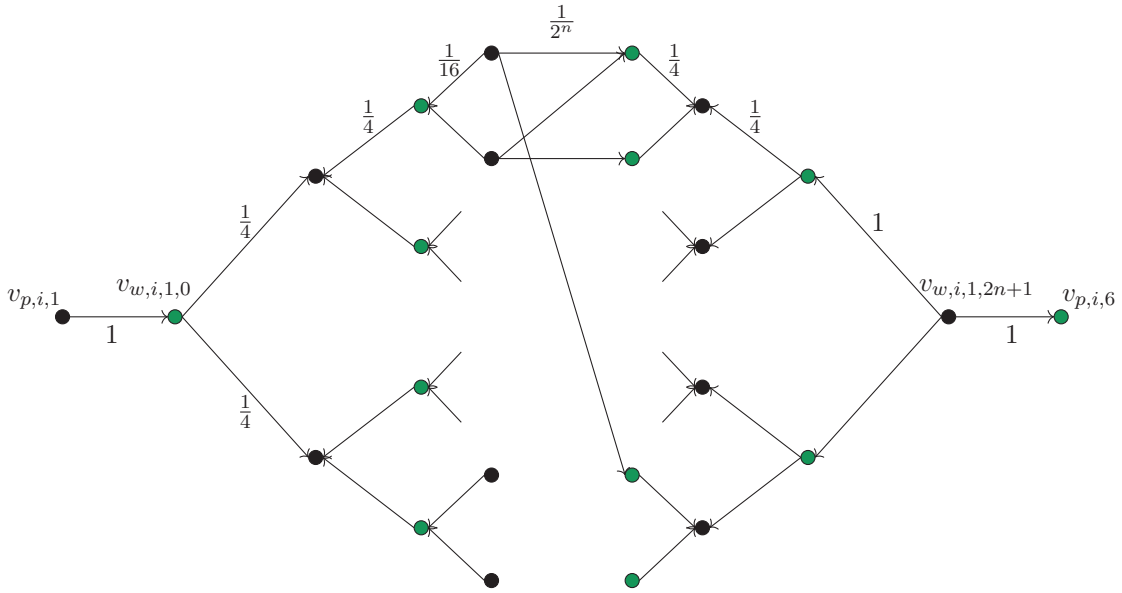


Figure 6.9: The 1st welded trees graph in the ith layer. For $j \in \{2,3\}$ the edge directions are simply reversed. The black vertices are the vertices in V_{even}, where the edge directions are reversed and where adjacent edges have the same weight and direction.

This setup induces the same partition of V into V_{even} and V_{odd} as in G_2 (visualised by coloured vertices in Figure 6.10). For each vertex $u \in V_{\text{even}}$, all adjacent edges have the same weight and direction, allowing us to easily generate the star state $|\psi_\star(u)\rangle$. For each $u \in V_{\text{odd}} \backslash \{s,t\}$, we have $|\psi_\star(u)\rangle \in \Psi_\star(u) = \widehat{\Psi}^3_\star(u)$.

All these names $v_{a,b,c}$ for referring to vertices are used purely for notation purposes to define the graph clearly. As in Section 6.3, we assign a random name from the set $\{0,1\}^{3n}$ to each vertex $u \in V$. To access the neighbours of a particular vertex, we are given quantum access to an adjacency list oracle O_G for the graph G. Given a $3n$-bit string $\sigma \in \{0,1\}^{3n}$ corresponding to a vertex $u \in V$, the adjacency list oracle O_G provides the bit strings of the neighbouring vertices in $\mathcal{N}(u)$. If σ does not correspond to any vertex, which is usually the case since $2^{3n} \gg |V|$, the oracle returns $\perp$.

Since the graph G consists of n identical subgraphs isomorphic to G_2, the analysis of the flow and potential vectors directly follows from Section 6.4.2. Beginning with the s-t alternative electrical flow θ^{alt}, we can construct this flow by simply connecting n s-t alternative electrical flows on each copy of G_2. This results in an alternative effective resistance $\mathcal{R}^{\mathsf{alt}}_{s,t} = \Theta(n^2)$. The alternative potential $\mathsf{p}^{\mathsf{alt}}$ can also be obtained directly by combining all the alternative potentials from each copy of G_2, where we add $((4+\mathcal{R})x + 3)(n-i)$ to each edge potential obtained from the copy of G_2 in the i-th layer. This ensures that for every $i \in [n-1]$,

$$|\mathsf{p}^{\mathsf{alt}}_{v_{p,i+1,1}}\rangle = ((4+\mathcal{R})x+3)(n-i)|\overline{\psi}_{v_{p,i+1,1}}\rangle,$$

implying $\mathsf{ET}^{\mathsf{alt}}_s = O(n^4)$. We now consider the following problem on the graph G, for which we present a quantum algorithm that can solve it exponentially faster than any classical algorithm.

Problem 6.4.1 (The pathfinding problem on a welded trees circuit graph)**.** *Given an adjacency list oracle O_G to the welded trees circuit graph G (as defined in Section 6.4.3) and the name of the starting vertex $s = 0^{3n}$, the goal is to output the names of the vertices on an s-t path.*

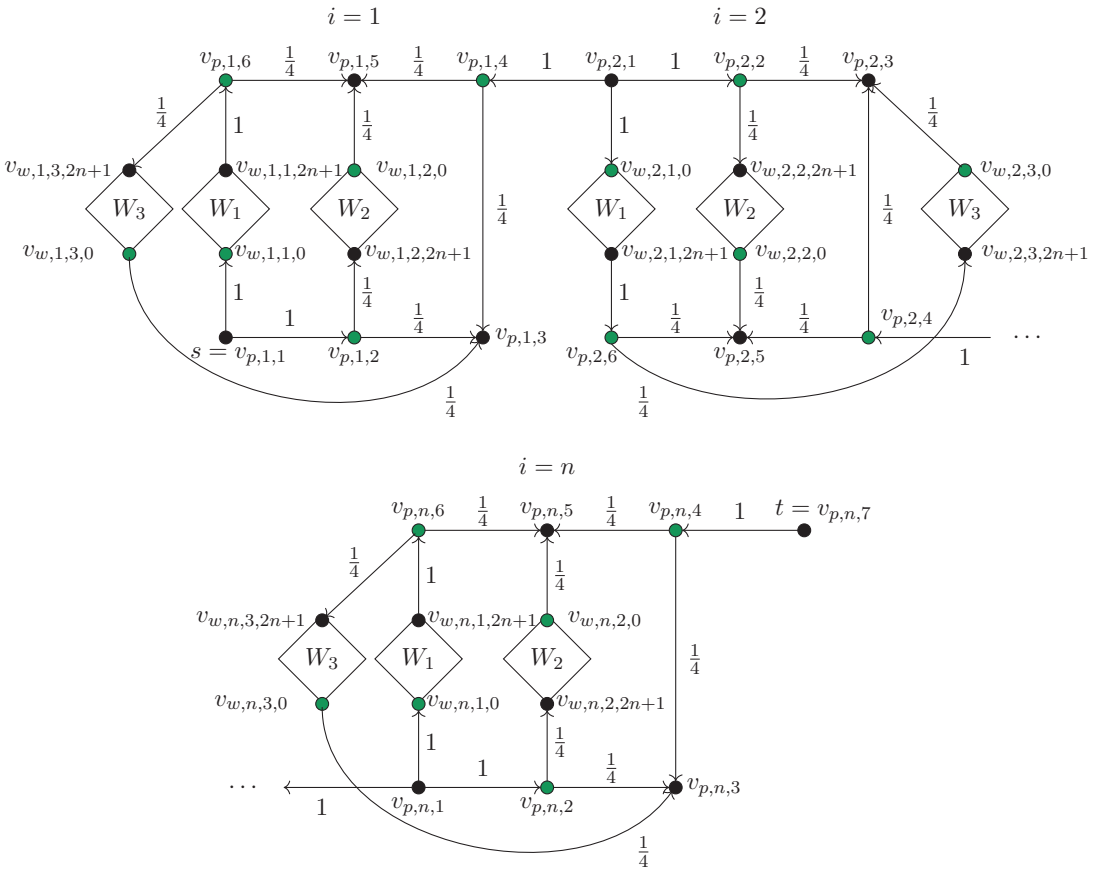


Figure 6.10: The welded trees circuit graph G showing all edge directions and edge weights. The coloured vertices are the vertices in V_{odd} and have the alternative neighbourhoods $\Psi_\star(u) = \widehat{\Psi}^3_\star(u)$ (see Definition 6.3.6). The black vertices are the vertices in V_{even}, where the edge directions are swapped and where adjacent edges have the same weight and direction. Each diamond, indexed by $j \in [3]$ represents the j'th welded trees graph in that layer. See Figure 6.9 for a detailed overview of the welded trees graph's structure.

6.4.4 The algorithm

In this section, we provide a quantum algorithm that can find the s-t shortest path in the welded trees circuit graph G, thereby solving Problem 6.4.1 in polynomial time.

Algorithm 2 Quantum algorithm for solving Problem 6.4.1

Require: Graph G as defined in Section 6.4.3, the starting vertex $s = 0^{3n}$, and a success probability parameter $\delta > 0$.
Ensure: The labels of an s-t path on G.

1. Set $i = 1$, $S = \emptyset$, and $T_1 = \Theta(n^2 \log(n))$.
2. Run phase estimation on the quantum walk operator $U_{\mathcal{A}^{\mathsf{alt}}\mathcal{B}}$ and state $|\psi_0\rangle = \frac{1}{\sqrt{2}}(|s_0, s\rangle + |s, s_0\rangle)$ to precision $O\left(\frac{\epsilon^2}{n^2}\right)$, where $\epsilon = O\left(\frac{1}{n^2}\right)$, and measure the phase register. If the output is "0", return the resulting state $|\widetilde{\theta}\rangle$.
3. Measure $|\widetilde{\theta}\rangle$ to obtain an outcome $|u, v\rangle$, representing the edge $(u, v) \in E$, and add it to S. If $i < T_1$, increment i by 1 and return to Step 2.
4. Use Breadth First Search to search through S for an s-t path and output the path if found.

Theorem 6.4.2. *Let the graph G be defined as in Section 6.4.3. Given an adjacency list oracle O_G for the graph G, there exists a quantum algorithm that solves Problem 6.4.1 with success probability $1 - O(\delta)$ and a cost of*

$$O\left(n^{10} \log(n) \log(n/\delta)\right) \text{ queries}, \qquad O\left(n^{11} \log(n) \log(n/\delta)\right) \text{ time}.$$

Proof. The proof involves analysing the cost and success probability of Algorithm 2, where we focus on the probability that the algorithm outputs the shortest path

$$\mathcal{P} = ((s, v_{p,1,2}), (v_{p,1,2}, v_{p,1,3}), (v_{p,1,3}, v_{p,1,4}), \ldots, (v_{p,n,4}, t)).$$

By Theorem 6.2.10, the phase estimation algorithm in Step 2 succeeds and returns an approximation of $|\theta^{\mathsf{alt}}\rangle$ with bounded error.

Suppose we had a perfect copy of $|\theta^{\mathsf{alt}}\rangle$. Measuring it would yield an edge $(u, v) \in \mathcal{P}$ with probability at least

$$\min_{(u,v)\in\mathcal{P}} \frac{1}{\mathcal{R}^{\mathsf{alt}}_{s,t}} \frac{(\theta^{\mathsf{alt}}_{u,v})^2}{\mathsf{w}_{u,v}} = \Omega\left(\frac{1}{n^2}\right).$$

Instead, we have access to a state $|\widetilde{\theta}\rangle$, which by Theorem 6.2.10 satisfies

$$\frac{1}{2}\left\| |\widetilde{\theta}\rangle\langle\widetilde{\theta}| - |\theta\rangle\langle\theta| \right\|_1 \leq \epsilon = O\left(\frac{1}{n^2}\right).$$

Hence, measuring $|\widetilde{\theta}\rangle$ yields an edge $(u, v) \in E$ containing the vertex t with probability at least $\Omega\left(\frac{1}{n^2}\right)$. The probability that one of the edges in $\mathcal{P}$ is not present in S after $T_1 = \Theta(n^2 \log(n))$ repetitions of Step 2 and Step 3 is therefore by the union bound at most:

$$|\mathcal{P}| \left(1 - O\left(\frac{1}{n^2}\right)\right)^{T_1} = O(1).$$

For the cost of Step 2, we require $O\left(n^4\right)$ calls to $U_{\mathcal{A}^{\mathsf{alt}}\mathcal{B}}$. By Lemma 6.3.8, each such call has a cost of $O(1)$ queries and $O(n)$ elementary operations. Since we can set up

the initial state $|\psi_0\rangle$ in $O(n)$ elementary operations (and no queries) and we run at most $T_1 = \Theta(n^2 \log(n))$ iterations of phase estimation, we find that the total contribution of Step 2 to the cost is

$$O(n^6 \log(n)) \text{ queries}, \qquad O(n^7 \log(n)) \text{ time}.$$

For Step 4, performing a Breadth First Search to find any s-t path in the subgraph defined by S requires $O(T_1) = O(n^2 \log(n/\delta))$ queries and basic operations. Thus, Step 2 dominates the overall cost. □

6.4.5 Classical lower bound

In this section, we demonstrate that our Algorithm 2 algorithm provides an exponential speedup compared to any classical algorithm, under the assumption that the following welded trees pathfinding problem is classically hard. To simplify the proof of our lower bound for the pathfinding problem Problem 6.4.1, we use the following assumption and the known classical lower bound for the welded trees problem.

Problem 6.4.3 (The welded trees pathfinding problem)**.** *Given an adjacency list oracle O_G for the welded trees graph G and the names of the starting vertex s and the ending vertex t, the objective is to output the names of the vertices along an s-t path.*

It is generally accepted that the welded trees pathfinding problem is difficult for classical algorithms, although no formal statement of this fact is widely available.

Assumption 6.4.4. *There exist constants $c_1 > 0$ and $c_2 \in (0, 2)$ such that any classical algorithm making at most $2^{n/6}$ queries to the oracle O_G for the welded trees graph G solves Problem 6.4.3 with probability at most $c_1 \cdot 2^{-c_2 n}$.*

Lemma 6.4.5 (Theorem 9 in [CCD+03])**.** *For the welded trees problem Problem 4.4.1, any classical algorithm making at most $2^{n/6}$ queries to the oracle O_G finds the ending vertex or a cycle with probability at most $4 \cdot 2^{-n/6}$.*

We follow the proof of the lower bound presented in [Li23], which builds on the lower bound proof from [CCD+03]. To establish the lower bound, we analyse the difficulty for any classical algorithm $\mathcal{A}$ to succeed in a simpler game:

Game A Let n be odd, and let G be the graph defined in Section 6.4.3. In Game A, a classical algorithm $\mathcal{A}$ wins if it outputs the name of the vertex $v_{p,(n+1)/2,1}$, or if the vertices visited by $\mathcal{A}$ contain a cycle. Following [CCD+03], we include the cycle condition to facilitate analysing the probability of $\mathcal{A}$ winning. This analysis involves determining whether a random embedding of a rooted binary tree into the random graph G contains either a cycle or the vertex $v_{p,(n+1)/2,1}$.

Given the starting vertex s, the random embedding of a rooted binary tree T into the graph G is defined as a function π from the vertices of T to the vertices of G, such that $\pi(\mathsf{ROOT}) = s$, and for any $(u, v) \in E(T)$, $\pi(u)$ and $\pi(v)$ are neighbours in G. We say that an embedding π is proper if $\pi(u) \neq \pi(v)$ for $u \neq v$. We say that T reaches its destination under π if $\pi(v) = v_{p,(n+1)/2,1}$. The random embedding can be generated as follows:

1. Set $\pi(\mathsf{ROOT}) = s$.

2. Let i and j be the two neighbours of ROOT in T, and let u and v be the neighbours of s in G. With probability $1/2$, set $\pi(i) = u$ and $\pi(j) = v$, otherwise set $\pi(i) = v$ and $\pi(j) = u$.

3. For any vertex i in T, if i is not a leaf and $\pi(i) \notin \{s, v_{p,(n+1)/2,1}\}$, let j and k denote the children of vertex i, and let ℓ denote its parent. Let u and v be the two neighbours of $\pi(i)$ in G other than $\pi(\ell)$. With probability $1/2$, set $\pi(j) = u$ and $\pi(k) = v$, otherwise set $\pi(j) = v$ and $\pi(k) = u$.

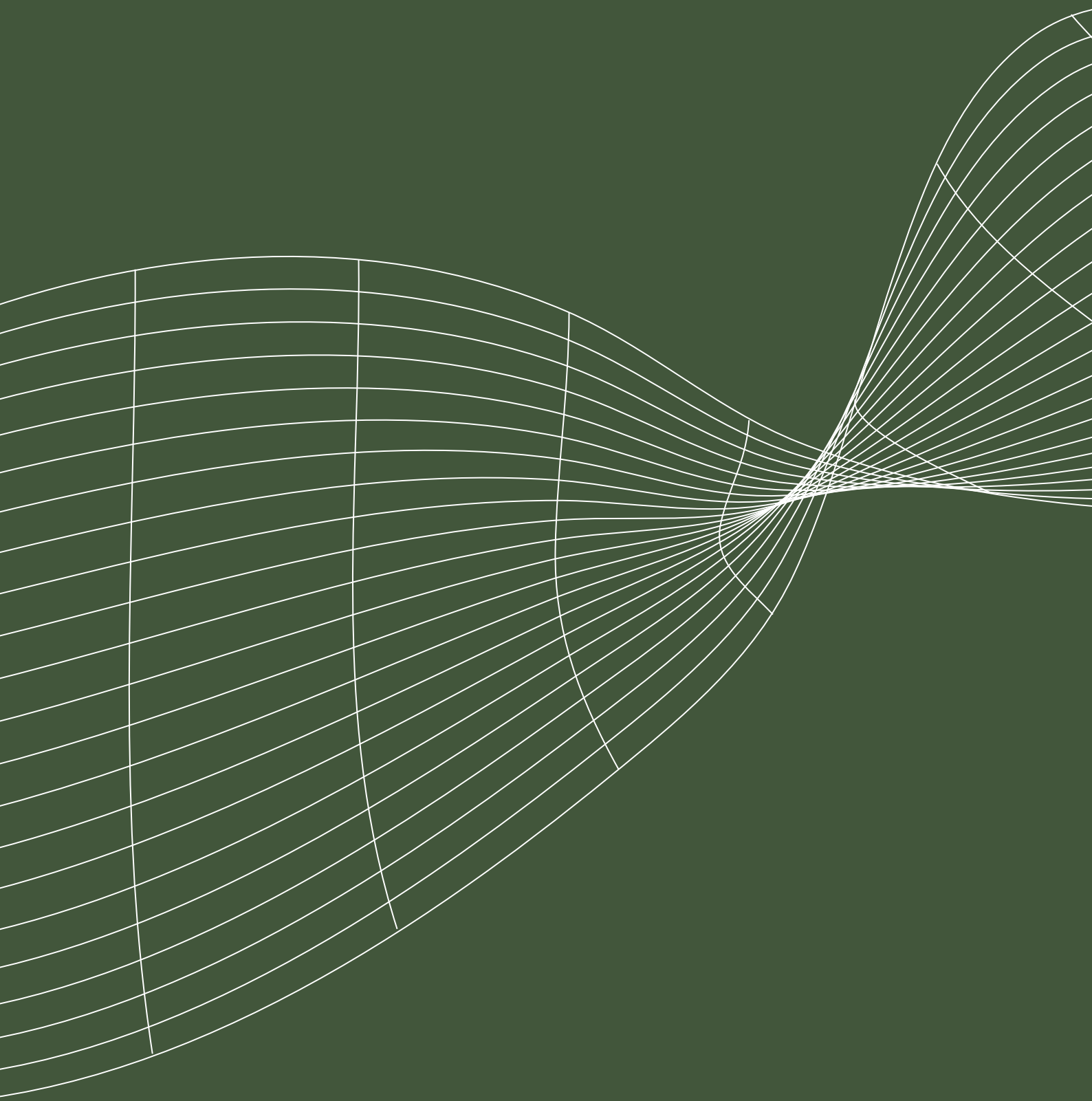
Theorem 6.4.6. *Let G be the graph defined in Section 6.4.3. Given the constants c_1 and c_2 from Assumption 6.4.4, and assuming that this assumption holds, any classical algorithm making at most $2^{n/6}$ queries to O_G solves Problem 6.4.1 with probability at most $(5 + c_1) \cdot 2^{-\min\{c_2, 1/6\}n}$.*

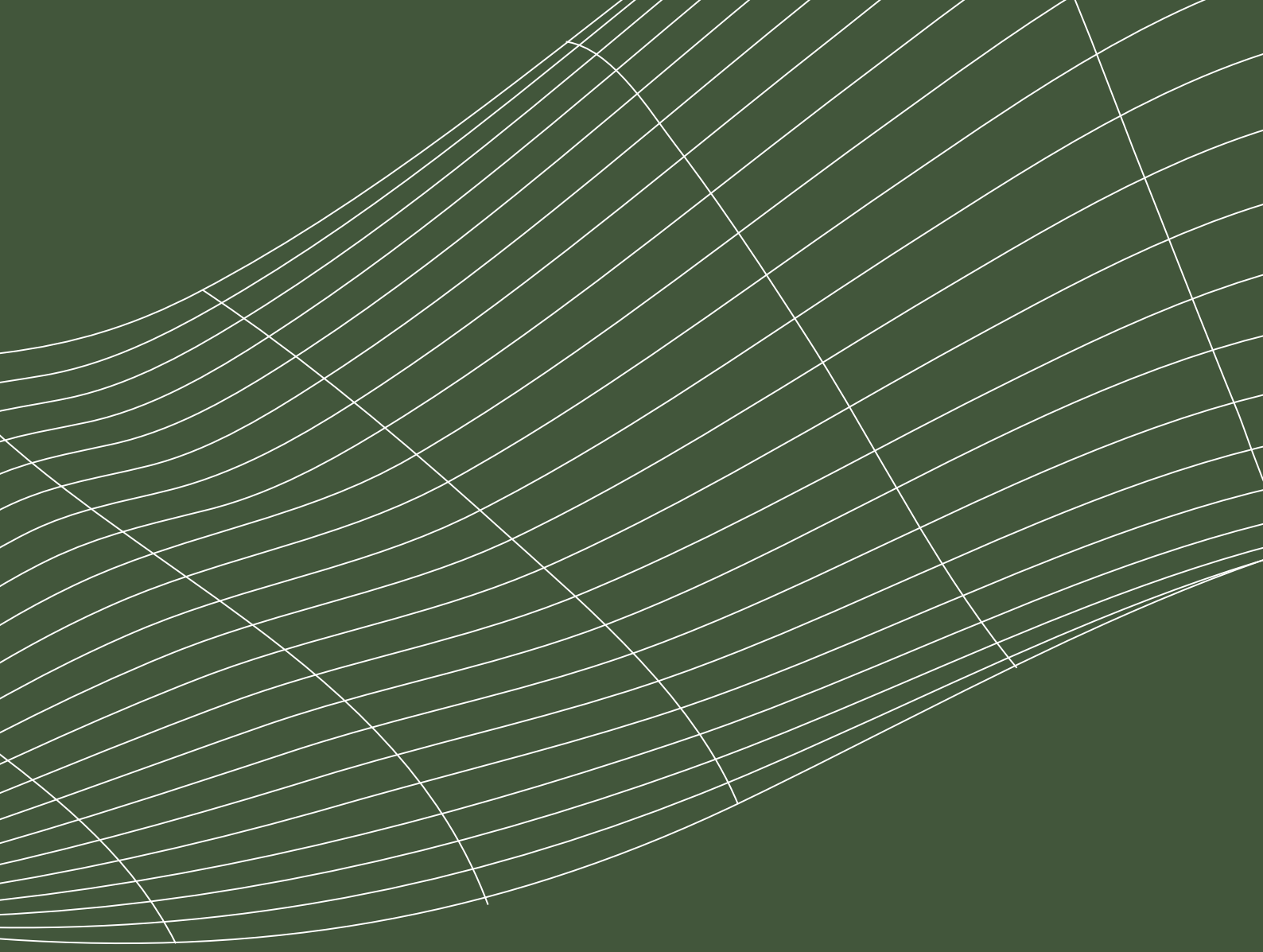
Proof. Let T be a random rooted binary tree with $2^{n/6}$ vertices, and let $\pi(T)$ denote its image in the graph G under the random embedding π. Given the starting vertex s, similar to [CCD+03], the probability of $\mathcal{A}$ winning Game A can be expressed as the probability that $\pi(T)$ contains either a cycle or the vertex $v_{p,(n+1)/2,1}$.

Firstly, $\mathcal{A}$ must enter a welded trees subgraph to find a cycle, as illustrated in Figure 6.10. There are two possibilities for finding a cycle in a welded trees subgraph. One is to find a cycle that includes only one root in one of the welded trees subgraphs. In this case, Lemma 6.4.5 states that, starting from one root, any classical algorithm making at most $2^{n/6}$ queries to the oracle finds the other root or a cycle with probability at most $4 \cdot 2^{-n/6}$. The other possibility is to find a cycle that includes two roots of a welded trees subgraph. According to Assumption 6.4.4, any classical algorithm making at most $2^{n/6}$ queries to the oracle finds such a cycle with probability at most $c_1 \cdot 2^{-c_2 n}$.

Now we assume that $\mathcal{A}$ does not encounter any cycles. Under this condition, the probability that $\mathcal{A}$ finds the vertex $v_{p,(n+1)/2,1}$ can be expressed as the probability that $\pi(T)$ reaches $v_{p,(n+1)/2,1}$, which requires π to follow the correct path $2n$ times, with probability 2^{-2n}. Given that there are at most $2^{n/6}$ attempts along each path of T, and there are at most $2^{n/6}$ paths, the probability of finding $v_{p,(n+1)/2,1}$ is at most $2^{n/3}2^{-2n} \leq 2^{-5n/3}$ by the union bound. The same result applies if the given name is t. Thus, the probability of $\mathcal{A}$ finding $v_{p,(n+1)/2,1}$ is at most $2 \cdot 2^{-5n/3}$.

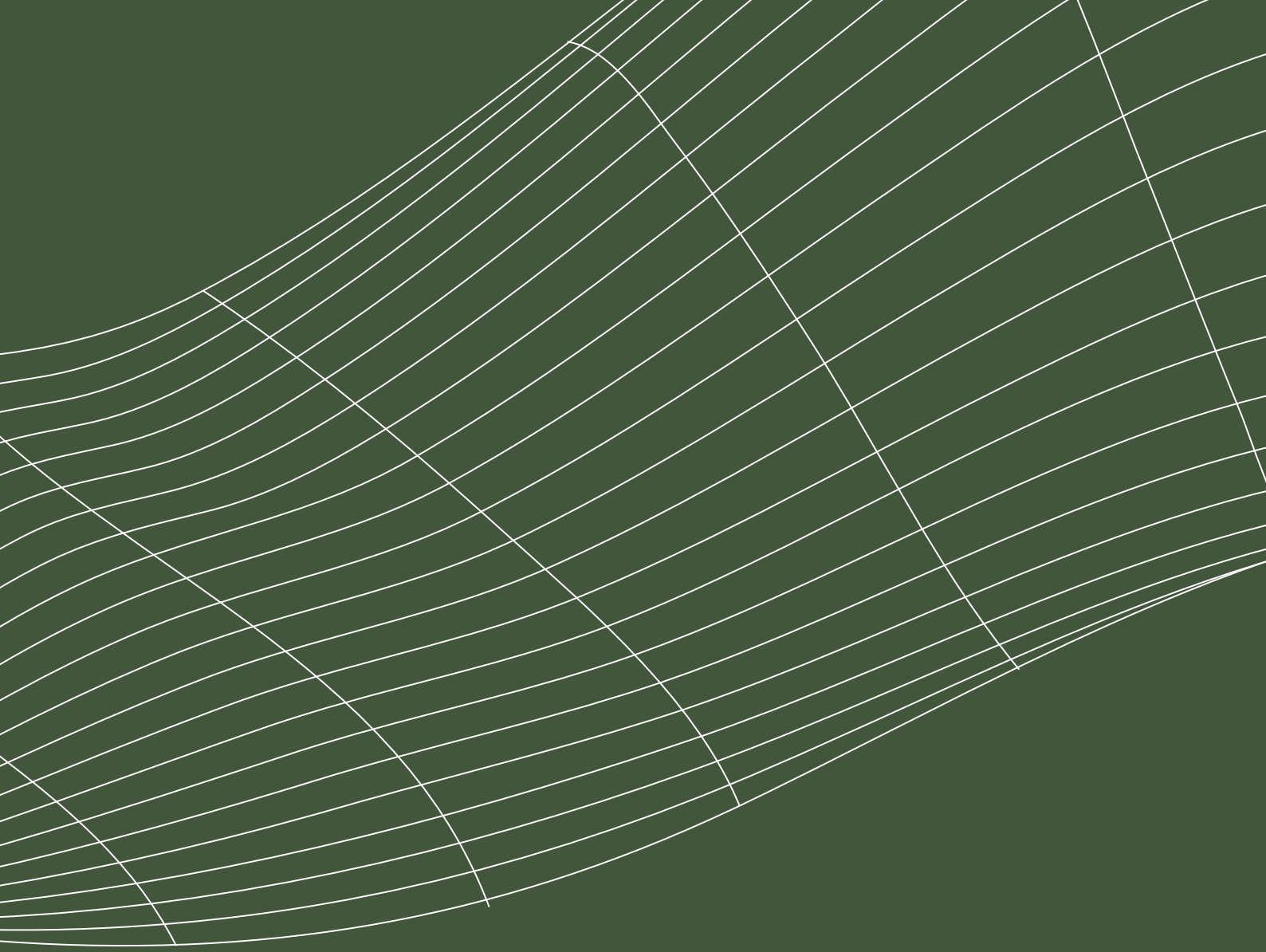
Combining these cases with the union bound gives a probability of at most $2^{-5n/3} + (4 + c_1) \cdot 2^{-\min\{c_2, 1/6\}n} \leq (5 + c_1) \cdot 2^{-\min\{c_2, 1/6\}n}$ for $\mathcal{A}$ winning Game A. Since solving Problem 6.4.1 guarantees a win in Game A, the theorem follows. □





Part II

Multiplicative Ladder Adversaries



Chapter 7

The compressed oracle is a worthy adversary

Lower bounds zijn heel simpel, maar het moeilijkste wat er is, zijn simpele lower bounds.

Johan Cruyff

This chapter is based on the paper *The Compressed Oracle is a Worthy Adversary* [JZ], which is joint work with Stacey Jeffery and is currently in preparation.

While it is fascinating to explore what quantum computers are capable of, it is equally important to study their limitations. One way to do this is by proving explicit lower bounds on the quantum query complexity of computational problems. In this chapter we study the compressed oracle technique [Zha19], which gives a framework to derive quantum query lower bounds for problems where a quantum algorithm interacts with a quantum oracle that accesses a random function. This framework was later formalised [CFHL21] and has proven to be very useful and convenient to use. It was unclear, however, how this frameworks compares with the more established lower bound methods for quantum query complexity. In this chapter, we answer this question by giving an explicit reduction to the multiplicative adversary method [Špa08]. To aid in this reduction, we introduce a simplification of the multiplicative adversary method that we call the multiplicative adversary ladder (MLA) method. This simplification is still powerful enough to encompass the compressed oracle technique and exhibit a strong direct product theorem. This could position the MLA method as a contender in the current quest to generalise the compressed oracle technique beyond non-product distributions. As further evidence of the potential of our MLA method, we demonstrate that it captures a recent approach by [Ros21].

7.1 Lower bounds and cryptography

Quantum query complexity lower bounds and the quantum security analysis of cryptographic protocols are two sides of the same coin, since proving security against a quantum adversary often requires upper-bounding the success probability of any quantum algorithm to solve a specific quantum query problem, in terms of the number of queries made by the adversary. The tools required for security proofs are generally stronger, since they should apply even for *average-case* inputs, and since it is usually necessary to show that the success probability with insufficient queries is much smaller than a constant, whereas quantum query lower bounds are most commonly in the bounded error setting.

7.1.1 Adversary methods

On the side of quantum query lower bounds, we have the original adversary method by [Amb00]. Applying this method reduces to mostly combinatorial arguments, which makes it very convenient to use, as shown by its many applications [BS04, DHHM06, BŠ06, DT07]. However, this method does have some technical limitations, one of which is the *certificate complexity barrier* [Zha05], which shows that there are problems for which this method cannot be tight. This limitation is addressed by the strictly stronger negative-weights adversary method [HLŠ07]. This method is, in fact, tight for the bounded-error regime for all quantum algorithms [Rei09], but this power comes at the cost of making it more complicated to use, as it lacks the primarily combinatorial reasoning present in the original adversary method. This means that even for very symmetric problems such as the *collision* problem [AS04], it is highly non-trivial to come up with an explicit lower bound resulting from the negative-weights adversary method, and the current construction relies on studying the symmetries of the problem via representation theory [BR17]. Moreover, the resulting lower bounds of the negative-weights adversary method are proportional to the algorithm's success probability, making them negligible for exponentially small success probabilities.

The latest and most powerful iteration in adversary methods is the multiplicative adversary method [Špa08], which is strictly stronger than the negative-weights adversary method [AMRR11]. Since the latter is already tight in the bounded-error case, this generalisation is particularly relevant in the low success probability regime, where it works even for exponentially small probabilities of success. This is a necessary condition for the method to exhibit a *strong direct product theorem* (SDPT), which intuitively states that to solve k independent instances of a function, one needs $\Omega(k)$ more queries to achieve better than an exponentially small (in k) probability of success. It was already shown in [Špa08] that the multiplicative adversary method satisfies a SDPT, which allowed [LR13] to prove a SDPT for quantum query complexity. Just as the negative-weights adversary method is more complex than the original adversary method, the multiplicative adversary method is much more complicated and, as a result, has not yet been used to provide novel quantum query lower bounds. This statement requires some nuance, however, as the multiplicative adversary method is a generalisation of an ad-hoc technique proposed in [Amb10a, AŠdW06] with the goal of showing a SDPT for symmetric Boolean functions.

For completeness, we also mention another ubiquitous quantum query lower bound technique, which predates the adversary bound: the polynomial method [BBC+01]. The multiplicative adversary method also includes the polynomial method as a special case, as shown in [MR15].

7.1.2 Compressed oracle technique

For cryptographic security proofs, quantum query lower bounds make little sense in the bounded-error regime. Ruling out adversaries that succeed with a high probability of success (at least $2/3$) is not enough, and it is necessary to rule out adversaries with very small probabilities as well. Moreover, average-case complexity is more relevant than worst-case complexity for this purpose: it is better to know that most cryptographic keys yield a secure construction than that there exists a cryptographic key such that the construction is secure. The compressed oracle technique [Zha19] does precisely this, as it yields an upper bound on the probability of success for any quantum algorithm interacting with a random oracle, giving an average-case complexity result that works even for exponentially small probabilities of success. Moreover, its analysis works via mostly combinatorial arguments, which makes it straightforward to apply and has quickly resulted in many results [LZ19a, CMSZ19, LZ19b, CFHL21, GHHM21, DFMS22]. It also satisfies a SDPT, allowing for quantum time-space tradeoffs [HM23]. The downside of this technique, however, is that it does not work for oracles that are not instantiated with a random function, or functions where each $f(x)$ is assigned independently [CMSZ19, HM23]. This is not only an issue for worst-case quantum query complexity but also rules out applications involving certain interesting cryptographic primitives such as random permutations.

7.1.3 Comparison

Both the adversary methods and the compressed oracle technique operate by tracking some progress that can only increase because each query establishes some amount of entanglement between the algorithm and the input. Since the adversary and compressed oracle techniques have different drawbacks that do not seem to exist in the other, it is interesting to see what the explicit relationship between these techniques is. This could aid in the ongoing search for a fusion of both techniques: a compressed oracle technique that can be applied to input distributions where each $f(x)$ is not necessarily assigned independently. On the cryptographic side, this could lead to quantum security proofs for schemes using random permutations, such as the sponge construction [BDPVA07]. On the quantum query lower bounds side, this might result in a technique that marries the power of the multiplicative adversary method—which works for all certificate complexities and exhibits a SDPT — with the pleasant intuitive combinatorial reasoning of the compressed oracle technique. Currently, the most promising result towards this “holy grail” has been a representation theory approach by [Ros21] that allows for tackling the problem of inverting a random permutation. This approach has recently been improved by [MMW24], where the algorithm has access to both a random permutation oracle and its inverse.

In this chapter, we demonstrate that a generalised compressed oracle technique — one that accommodates distributions beyond random functions and permutations — must fall somewhere between the compressed oracle technique and the multiplicative adversary method. We explicitly show this by proving that the compressed oracle technique reduces to the multiplicative adversary method. This implies that any lower bound obtained by the compressed oracle technique can also be achieved through the multiplicative adversary method. We achieve this by constructing a weaker version of the multiplicative adversary method, which still satisfies a strong direct product theorem (SDPT) and remains more powerful than the compressed oracle technique. We hope that this intermediate version will aid in the search for an extended compressed oracle technique, as we show that it incorporates the approach from [Ros21] as a special case.

7.1.4 Quantum query complexity

In the quantum query model (see Section 2.2.5), we are generally interested in computing a function $\mathcal{F} : \mathsf{Func} \to \Sigma$ on an input $f \in \mathsf{Func}$. We consider the case where Func is a subset of Y^X, so each f can itself also be viewed as a function from X to Y. For example, if $Y = \{0, 1\}$ and $X = [n]$, then $f \in \mathsf{Func}$ is an n-bit string (which might have a promise defined by the subset Func). In this chapter, we restrict ourselves to X being any finite set of size N and consider Y to be the finite set $[M-1]_0$.

The memory of our quantum algorithm $\mathcal{A}$, tasked with computing $\mathcal{F}$ on an input f, is described without loss of generality by the registers $\mathcal{W}$, $\mathcal{X}$, and $\mathcal{Y}$. Here, the input oracle acts on $\mathcal{X} \times \mathcal{Y}$ (as detailed below), while $\mathcal{W}$ represents an additional workspace. The input function $f \in \mathsf{Func}$ can be accessed by $\mathcal{A}$ via an oracle, defined as follows:

Definition 7.1.1 (Oracle)**.** *Fix a finite set X of size N and let $Y = [M-1]_0$. An* oracle $\mathcal{O}_f$*, encoding the input function $f \in \mathsf{Func}$, is a unitary transformation that acts on*

$$\mathrm{span}\{|x\rangle_{\mathcal{X}}|y\rangle_{\mathcal{Y}} : x \in X, y \in Y\},$$

with its action on the basis state $|x\rangle_{\mathcal{X}}|y\rangle_{\mathcal{Y}}$ defined as

$$\mathcal{O}_f|x\rangle_{\mathcal{X}}|y\rangle_{\mathcal{Y}} = |x\rangle_{\mathcal{X}}|(y + f(x)) \bmod M\rangle_{\mathcal{Y}}.$$

The input f is typically drawn from some (hard) input distribution σ over Func, denoted $f \sim \sigma$. Consequently, $\mathcal{O}_f$ is a random variable. In adversary methods and the compressed oracle technique, this randomness is avoided by introducing an additional register $\mathcal{I}$, which stores a superposition of function tables representing the input f. In quantum information theory, this is known as *purification*. If $f \sim \sigma$, the register $\mathcal{I}$ will be initialised as

$$|\sigma\rangle = \sum_{f \in Y^X} \sqrt{\sigma(f)}|f\rangle_{\mathcal{I}}.$$

Here, $|\sigma\rangle$ represents the initial state of the input register. It is important to note that this should not be confused with the initial state of the algorithm, which is the all-zero state. In the adversary frameworks literature, the initial state of the input register is often denoted by $|\delta\rangle$. This purification of the input leads to the following purified oracle:

Definition 7.1.2 (Purified Oracle)**.** *Fix a finite set X of size N and let $Y = [M-1]_0$. A* purified oracle $\mathcal{O}$ *is a unitary transformation that acts on*

$$\mathrm{span}\{|x\rangle_{\mathcal{X}}|y\rangle_{\mathcal{Y}}|f\rangle_{\mathcal{I}} : x \in X, y \in Y, f \in Y^X\},$$

with its action on the basis state $|x\rangle_{\mathcal{X}}|y\rangle_{\mathcal{Y}}|f\rangle_{\mathcal{I}}$ defined as

$$\mathcal{O}|x\rangle_{\mathcal{X}}|y\rangle_{\mathcal{Y}}|f\rangle_{\mathcal{I}} = |x\rangle_{\mathcal{X}}|(y + f(x)) \bmod M\rangle_{\mathcal{Y}}|f\rangle_{\mathcal{I}}.$$

From the perspective of the algorithm, it is indistinguishable whether it interacts with the random variable $\mathcal{O}_f$ or the purified oracle $\mathcal{O}$ with input register initialised to $|\sigma\rangle$. The relationship between the two is captured by the following expression:

$$\mathcal{O} = \sum_{f \in Y^X} \mathcal{O}_f \otimes |f\rangle\langle f|_{\mathcal{I}}.$$

It is equivalent, and in this work more convenient, to encode the query into the phase by viewing the $\mathcal{Y}$ register in the Fourier basis $\{|\hat{y}\rangle\}_{y \in Y}$ (see Definition 2.2.1) instead of the computational basis $\{|y\rangle\}_{y \in Y}$. In this Fourier basis, the oracle from Definition 7.1.2 acts on any basis state $|x\rangle_{\mathcal{X}}|\hat{y}\rangle_{\mathcal{Y}}|f\rangle_{\mathcal{I}}$ as

$$\mathcal{O}|x\rangle_{\mathcal{X}}|\hat{y}\rangle_{\mathcal{Y}}|f\rangle_{\mathcal{I}} = e^{\frac{2\pi\iota}{M} y f(x)}|x\rangle_{\mathcal{X}}|\hat{y}\rangle_{\mathcal{Y}}|f\rangle_{\mathcal{I}}.$$

Additionally, it will often be convenient to decompose the oracle $\mathcal{O}$ into diagonal unitary matrices $\mathcal{O}_{x,y}$ given by

$$\mathcal{O} = \sum_{x \in X, y \in Y} |x\rangle\langle x|_{\mathcal{X}} \otimes |\hat{y}\rangle\langle\hat{y}|_{\mathcal{Y}} \otimes \mathcal{O}_{x,y}, \tag{7.1}$$

where each $\mathcal{O}_{x,y}$ acts on the basis state $|f\rangle_{\mathcal{I}}$ as

$$\mathcal{O}_{x,y}|f\rangle_{\mathcal{I}} = e^{\frac{2\pi\iota}{M} y \cdot f(x)} |f\rangle_{\mathcal{I}}.$$

Definition 7.1.3 (T-Query Quantum Algorithm)**.** *Fix a set X of size N and let $Y = [M-1]_0$. A T-*query quantum algorithm *$\mathcal{A}$ on Y^X is a sequence of unitaries $U_0, \ldots, U_T$ on*

$$\mathrm{span}\{|w\rangle_{\mathcal{W}}|x\rangle_{\mathcal{X}}|y\rangle_{\mathcal{Y}} : w \in W, x \in X, y \in Y\},$$

for some finite set W. For a fixed algorithm $\mathcal{A}$ and a fixed input distribution σ, let

$$|\sigma\rangle = \sum_{f \in Y^X} \sqrt{\sigma(f)} |f\rangle_{\mathcal{I}},$$

and let

$$|\psi_t(\mathcal{A}, \sigma)\rangle = U_t \mathcal{O} U_{t-1} \mathcal{O} \ldots \mathcal{O} U_0 |0\rangle_{\mathcal{W}\mathcal{X}\mathcal{Y}} |\sigma\rangle_{\mathcal{I}}$$

denote the state of the algorithm before the $(t+1)$-th query is made, and let

$$\rho_{\mathcal{I}}^t(\mathcal{A}, \sigma) = \mathrm{Tr}_{\mathcal{W}\mathcal{X}\mathcal{Y}} \left[|\psi_t(\mathcal{A}, \sigma)\rangle\langle\psi_t(\mathcal{A}, \sigma)| \right]$$

denote the reduced state of the input register (see (2.5)), which we call the input register states for $\mathcal{A}$ and $|\sigma\rangle$. *When $\mathcal{A}$ and $|\sigma\rangle$ are clear from context, we will omit the $(\mathcal{A}, \sigma)$ notation.*

In the definition of $|\psi_t(\mathcal{A}, \sigma)\rangle$, both the queries $\mathcal{O}$ and the unitaries $U_1, \ldots, U_t$ act on a larger Hilbert space than originally defined, but recall from Section 2.2.1 that each operator is implicitly understood to act tensored with the identity operator on any unaffected registers.

In this chapter, we compare two methods designed to lower bound the quantum query complexity of a problem $\mathcal{F}$:

Definition 7.1.4 (ϵ-error Quantum Query Complexity)**.** *Fix $\mathcal{F} : \mathsf{Func} \to \Sigma$. Then the* ϵ-error quantum query complexity *of $\mathcal{F}$, denoted by $Q_{1-\epsilon}(\mathcal{F})$, is the minimum number of queries needed by any quantum query algorithm $\mathcal{A}$ to successfully output $\mathcal{F}(f)$ for every input $f \in \mathsf{Func}$ with success probability at least $1 - \epsilon$.*

7.2 The frameworks

In this section, we introduce the two lower bound frameworks that will be compared throughout this chapter: the multiplicative adversary method and the compressed oracle technique.

7.2.1 The multiplicative adversary method

The general idea behind the adversary methods is that any algorithm for $\mathcal{F}$, run on a superposition of different inputs $|\sigma\rangle$ with different values of $\mathcal{F}$, must entangle the algorithm's workspace $\mathcal{W}\mathcal{X}\mathcal{Y}$ (which must eventually contain the answer) with the input register $\mathcal{I}$, resulting in the reduced density matrix on $\mathcal{I}$, which is initially the pure state $\rho_{\mathcal{I}}^0(\mathcal{A}, \sigma) = |\sigma\rangle\langle\sigma|$, becoming some mixed state $\rho_{\mathcal{I}}^T(\mathcal{A}, \sigma)$.

This idea was already present in the original *quantum adversary method* [Amb00], which was later generalised to the stronger *negative-weights adversary method* [HLŠ07] (now often called the adversary method), which is tight in the bounded-error regime, i.e. $\epsilon \leq 1/3$. We will be interested in the even more powerful *multiplicative adversary method*, first formalised in [Špa08] and further developed in [AMRR11]. We now describe this method.

Definition 7.2.1 (Multiplicative Adversary Matrix)**.** *Fix $\mathcal{F} : \mathsf{Func} \to \Sigma$. A* multiplicative adversary matrix *for problem $\mathcal{F}$ is a positive definite matrix $\Gamma \in \mathbb{C}^{\mathsf{Func}\times\mathsf{Func}}$ with smallest eigenvalue 1.*

Any multiplicative adversary matrix gives rise to a *progress measure*, which is a way of quantifying how much progress a quantum algorithm $\mathcal{A}$ has made after t queries towards solving a particular problem $\mathcal{F}$.

Definition 7.2.2 (Progress)**.** *Fix a problem $\mathcal{F} : \mathsf{Func} \to \Sigma$, and input distribution σ supported on* Func*. Fix a multiplicative adversary matrix Γ for $\mathcal{F}$, as in Definition 7.2.1, with 1-eigenstate $|\sigma\rangle$ and a T-query quantum algorithm $\mathcal{A}$, as in Definition 7.1.3. Let $\rho^t_{\mathcal{I}}(\mathcal{A}, \sigma)$ be the input register states for $\mathcal{A}$ and input distribution σ before the $(t+1)$-th query is made. The associated* progress measure *for $t \in [T]_0$ is defined as*

$$W^t(\Gamma, \mathcal{A}) := \mathrm{Tr}[\Gamma \rho^t_{\mathcal{I}}(\mathcal{A}, \sigma)].$$

Theorem 7.2.3 quantifies in what way we can think of $W^t(\Gamma, \mathcal{A})$ as a "progress measure." After 0 queries, we have made no progress, which is indicated by $W^0(\Gamma, \mathcal{A}) = 1$ (Item 1). After T queries, if we want to claim that the algorithm actually solves $\mathcal{F}$ with probability $1 - \epsilon$, then it must be the case that the progress $W^T(\Gamma, \mathcal{A})$ has increased sufficiently above 1 (Item 3). Item 2 bounds the amount of progress that can be made in a single query.

Theorem 7.2.3 ([Špa08, AMRR11])**.** *Fix a problem $\mathcal{F} : \mathsf{Func} \to \Sigma$, an input distribution σ on* Func*, and a multiplicative adversary matrix Γ for $\mathcal{F}$ with 1-eigenstate $|\sigma\rangle$. Let λ be a real number with $1 < \lambda \leq \|\Gamma\|$. Let Λ_{bad} be the projector onto the eigenspaces of Γ corresponding to eigenvalues smaller than λ and let $\eta \leq 1-\epsilon$ be a positive constant such that $\|F_z \Lambda_{\mathsf{bad}}\|^2 \leq \eta$ for every $z \in \Sigma$, where $F_z = \sum_{\substack{f\in\mathsf{Func}:\\ \mathcal{F}(f)=z}} |f\rangle\langle f|$. Then:*

1. *For any quantum algorithm $\mathcal{A}$, $W^0(\Gamma, \mathcal{A}) = 1$.*
2. *For any T-query quantum algorithm $\mathcal{A}$, and $t \in [T-1]_0$,*
$$\frac{W^{t+1}(\Gamma, \mathcal{A})}{W^t(\Gamma, \mathcal{A})} \leq \max_{x\in X, y\in Y} \left\| \mathcal{O}^\dagger_{x,y} \Gamma^{1/2} \mathcal{O}_{x,y} \Gamma^{-1/2} \right\|^2.$$
3. *For any T-query quantum algorithm $\mathcal{A}$ that solves $\mathcal{F}$ on input $|\sigma\rangle$ with success probability at least $1-\epsilon$, $W^T(\Gamma, \mathcal{A}) \geq 1 + (\lambda - 1)\left(\sqrt{1-\epsilon} - \sqrt{\eta}\right)^2$.*

Corollary 7.2.4. *For any η that satisfies the constraints of Theorem 7.2.3, $\epsilon \in (0, 1-\eta)$, problem $\mathcal{F} : \mathsf{Func} \to \Sigma$, and input distribution σ on* Func*,*

$$Q_{1-\epsilon}(\mathcal{F}) \geq \max_{\Gamma,\lambda} \frac{\log\left(1 + (\lambda-1)\left(\sqrt{1-\epsilon} - \sqrt{\eta}\right)^2\right)}{\log\left(\max_{x\in X, y\in Y} \left\| \mathcal{O}^\dagger_{x,y} \Gamma^{1/2} \mathcal{O}_{x,y} \Gamma^{-1/2} \right\|^2\right)},$$

where Γ ranges over all multiplicative adversary matrices for $\mathcal{F}$ with 1-eigenstate $|\sigma\rangle$ (see Definition 7.2.1) and λ ranges over $[1, \|\Gamma\|]$.

7.2.2 Dealing with search problems

By Definition 7.1.4, we aim to lower bound the number of queries that any quantum query algorithm makes to successfully output $\mathcal{F}(f) \in \Sigma$ for any input $f \in \mathsf{Func}$. All *decision* problems can be phrased in this form, where the set Σ is equal to $\{0,1\}$. However, it is not always possible to interpret more general *search* problems as computing a single-valued function $\mathcal{F}(f)$.

For instance, consider the simplest search problem, known as *Search* (see Section 2.2.5). In this case, all goes well: we have $f \in \mathsf{Func}$ as an n-bit string with Hamming weight 1, and $\mathcal{F}(f)$ is defined as the unique index i such that $f(i) = 1$, implying $\Sigma = [n]$. However, if we relax Func to include all n-bit strings with Hamming weight at least 1, then there are multiple correct indices i such that $f(i) = 1$. Consequently, there is no longer a single correct value for $\mathcal{F}(f)$ for each $f \in \mathsf{Func}$. Further generalising Func to include all n-bit strings leads to cases where some inputs contain no indices mapping to 1, making $\mathcal{F}(f)$ undefined for such inputs.

In search problems, the problem is therefore characterised by a *relation* $\mathcal{R} \subset \mathsf{Func} \times \Sigma$, and the algorithm must output some $\sigma \in \Sigma$ on input f such that $(f, \sigma) \in \mathcal{R}$. This formulation generalises the concept of computing a function $\mathcal{F}$, as we can define the relation $\mathcal{R}$ corresponding to $\mathcal{F}$ as the set $\{(f, \mathcal{F}(f)) : f \in \mathsf{Func}\}$. We shall see in Theorem 7.2.7 that the compressed oracle framework solves such search problems.

However, to remain closer to the notation used in Theorem 7.3.4, we still choose to frame search problems in terms of computing a function $\mathcal{F}$. To accommodate the fact that search problems can have multiple, or even no, correct outputs, we define that a quantum query algorithm $\mathcal{A}$ has successfully computed a function $\mathcal{F}$ on an input $f \in \mathsf{Func}$ if it outputs z such that $z \in \mathcal{F}(f)$. Consequently, if Σ is the set of possible outputs, then each $\mathcal{F}(f)$ is a subset of Σ. To distinguish this from the earlier case where $\mathcal{F}(f)$ is a single value, we now write $\mathcal{F} : \mathsf{Func} \to 2^{\Sigma}$ for search problems.

To reflect the modified definition of "success", we also update the projector F_z for each $z \in \Sigma$ in Theorem 7.2.3 to:

$$F_z = \sum_{\substack{f \in \mathsf{Func}: \\ \mathcal{F}(f) \ni z}} |f\rangle\langle f|.$$

We show that these modifications do not impact Item 3 in Theorem 7.2.3, thereby generalising Theorem 7.2.3 to search problems:

Lemma 7.2.5. *Let Γ be a multiplicative adversary matrix for a problem $\mathcal{F} : \mathsf{Func} \to 2^{\Sigma}$ and let λ satisfy the constraints of Theorem 7.2.3. Let Λ_{bad} be the projector onto the eigenspaces of Γ corresponding to eigenvalues smaller than λ and let $\eta \leq 1 - \epsilon$ be a positive constant such that $\|F_z \Lambda_{\mathsf{bad}}\|^2 \leq \eta$ for every $z \in \Sigma$, where $F_z = \sum_{\substack{f \in \mathsf{Func}: \\ \mathcal{F}(f) \ni z}} |f\rangle\langle f|$.*

Then for any T-query quantum algorithm $\mathcal{A}$ that solves $\mathcal{F}$ on input $|\sigma\rangle$ with success probability at least $1 - \epsilon$,

$$W^T(\Gamma, \mathcal{A}) \geq 1 + (\lambda - 1)\left(\sqrt{1-\epsilon} - \sqrt{\eta}\right)^2.$$

Proof. Consider the final state $|\psi_T(\mathcal{A}, \sigma)\rangle$ at the end of the computation. The output is correct if and only if $z \in \mathcal{F}(f)$, meaning we can define a "success" measurement on the input register $\mathcal{I}$ and the workspace register $\mathcal{W}_O$ containing the output $z \in \Sigma$:

$$\Lambda_{\mathsf{succ}} := \sum_{z \in \Sigma} |z\rangle\langle z|_{\mathcal{W}_O} \otimes F_z.$$

Since the algorithm $\mathcal{A}$ solves $\mathcal{F}$ with success probability at least $1 - \epsilon$ on the input $|\sigma\rangle$, we know that

$$\|\Lambda_{\mathsf{succ}} |\psi_T(\mathcal{A}, \sigma)\rangle\| \geq \sqrt{1-\epsilon}. \tag{7.2}$$

As in the original proof of Item 3 in [Špa08], we define $\Lambda_{\mathsf{good}} := I - \Lambda_{\mathsf{bad}}$ as the projector onto the orthogonal complement of the bad subspace, which we call the good subspace. Using these projectors, we decompose $|\psi_T(\mathcal{A},\sigma)\rangle$ as follows:

$$|\psi_T(\mathcal{A},\sigma)\rangle = \sqrt{1-\beta}|\Psi_{\mathsf{bad}}\rangle + \sqrt{\beta}|\Psi_{\mathsf{good}}\rangle, \tag{7.3}$$

where

$$|\Psi_{\mathsf{bad}}\rangle = \frac{\Lambda_{\mathsf{bad}}|\psi_T(\mathcal{A},\sigma)\rangle}{\|\Lambda_{\mathsf{bad}}|\psi_T(\mathcal{A},\sigma)\rangle\|}, \quad |\Psi_{\mathsf{good}}\rangle = \frac{\Lambda_{\mathsf{good}}|\psi_T(\mathcal{A},\sigma)\rangle}{\|\Lambda_{\mathsf{good}}|\psi_T(\mathcal{A},\sigma)\rangle\|}, \text{ and } \quad \beta = \|\Lambda_{\mathsf{good}}|\psi_T(\mathcal{A},\sigma)\rangle\|^2 .$$

We proceed by separately bounding the contributions of the "good" and "bad" components to $\|\Lambda_{\mathsf{succ}}|\psi_T(\mathcal{A},\sigma)\rangle\|$. For the "good" component, we can use the trivial bound, namely $\|\Lambda_{\mathsf{succ}}|\Psi_{\mathsf{good}}\rangle\| \leq 1$. For the "bad" component, we bound it by

$$\|\Lambda_{\mathsf{succ}}|\Psi_{\mathsf{bad}}\rangle\| \leq \max_{z\in\Sigma} \|F_z\Lambda_{\mathsf{bad}}\| \leq \sqrt{\eta}.$$

Combining this with (7.3) and (7.2), we find that

$$\sqrt{1-\epsilon} \leq \|\Lambda_{\mathsf{succ}}|\psi_T(\mathcal{A},\sigma)\rangle\| \leq \sqrt{1-\beta}\,\|\Lambda_{\mathsf{succ}}|\Psi_{\mathsf{bad}}\rangle\| + \sqrt{\beta}\,\|\Lambda_{\mathsf{succ}}|\Psi_{\mathsf{good}}\rangle\| \leq \sqrt{\eta} + \sqrt{\beta},$$

which we can rearrange to obtain $\beta \geq \left(\sqrt{1-\epsilon} - \sqrt{\eta}\right)^2$.

Having found a lower bound on β, we can now apply the same decomposition from (7.3) to our progress measure to conclude the lemma:

$$\begin{aligned} W^T(\Gamma,\mathcal{A}) = \mathrm{Tr}(\Gamma\rho^T_{\mathcal{A}}(\mathcal{A},\sigma)) &\geq \mathrm{Tr}(\lambda\Lambda_{\mathsf{good}}\rho^T_{\mathcal{A}}(\mathcal{A},\sigma)) + \mathrm{Tr}(\Lambda_{\mathsf{bad}}\rho^T_{\mathcal{A}}(\mathcal{A},\sigma)) \\ &\geq \lambda\beta + (1-\beta) \geq 1 + (\lambda-1)\left(\sqrt{1-\epsilon} - \sqrt{\eta}\right)^2 . \end{aligned}$$ □

7.2.3 The compressed oracle technique

In the compressed oracle technique [Zha19], Zhandry observes that in query problems where the algorithm interacts with a quantum random oracle, it is equivalent (by applying a purification) to assume that the algorithm is run on a uniform superposition over all possible functions from the set X to the set Y. In this picture, a quantum adversary interacting with the quantum random oracle towards some nefarious end is analogous to a quantum algorithm run on input distribution σ, which is initialised to the uniform distribution over all functions from X to Y:

$$|\mathsf{Uniform}\rangle_{\mathcal{I}} := \frac{1}{\sqrt{M^N}} \sum_{f\in Y^X} |f\rangle_{\mathcal{I}}. \tag{7.4}$$

We refrain from discussing the compressed oracle in depth here. For more details, see [Zha19, CMSZ19, CFHL21]. Instead, we summarise the necessary parts needed to show how the compressed oracle technique can be used to derive quantum query lower bounds whenever $\mathsf{Func} = Y^X$. The input register $\mathcal{I}$ holding any computational basis state $|f\rangle_{\mathcal{I}}$, where $f \in Y^X$, can be viewed as a tensor product of the different function values for f for different values of $x \in X$:

$$|f\rangle_{\mathcal{I}} = \bigotimes_{x\in X} |f(x)\rangle_{\mathcal{I}_x}.$$

This can be interpreted as a look-up table that fully describes the action of f. We can also consider a Fourier basis (see Definition 2.2.1) for this register that represents a

function in Y^X. Let $\{|\hat{f}\rangle\}_{f\in Y^X}$ be the *Fourier basis* of $\mathcal{I}\equiv\mathcal{Y}^{\otimes N}$, where each $|\hat{f}\rangle$ is defined as

$$|\hat{f}\rangle_{\mathcal{I}} \coloneqq \bigotimes_{x\in X} \mathsf{QFT}_M|f(x)\rangle_{\mathcal{I}_x} = \bigotimes_{x\in X} |\widehat{f(x)}\rangle_{\mathcal{I}_x}.$$

From the look-up table perspective, this means that we change the basis of all our entries in the look-up table. The key insight that Zhandry makes is that if we view both the input register $\mathcal{I}$ in this Fourier basis, as well as the $\mathcal{Y}$ register, then a query (as in Definition 7.1.2) acts on a basis state $|x\rangle_{\mathcal{X}}|\hat{y}\rangle_{\mathcal{Y}}|\hat{f}\rangle_{\mathcal{I}}$ as follows:

$$\mathcal{O}\left(|x\rangle_{\mathcal{X}}|\hat{y}\rangle_{\mathcal{Y}}|\hat{f}\rangle_{\mathcal{I}}\right) = |x\rangle_{\mathcal{X}}|\hat{y}\rangle_{\mathcal{Y}}|\widehat{f-y\cdot\delta_x}\rangle_{\mathcal{I}}. \tag{7.5}$$

Here, δ_x denotes the point function satisfying $\delta_x(x)=1$ and $\delta_x(x')=0$ for all $x'\neq x$. This change of perspective is quite peculiar: where in a regular query (as in Definition 7.1.2) the information stored in the $\mathcal{I}_x$ register is "copied" into the $\mathcal{Y}$ register, this interaction is mirrored when viewing the $\mathcal{I}_x$ register in the Fourier basis. Another added benefit of this basis change is that the initial state $|\mathsf{Uniform}\rangle$ simplifies to

$$\mathsf{QFT}_M^{\otimes N}|\mathsf{Uniform}\rangle_{\mathcal{I}} = \bigotimes_{x\in X}|\hat{0}\rangle_{\mathcal{I}_x}. \tag{7.6}$$

The action of the oracle in (7.5), combined with (7.6), implies the following consequence, which is the cornerstone of the compressed oracle technique:

Fact 7.2.6. *For any T-query quantum algorithm $\mathcal{A}$ and for any $t\in[T]_0$, we have that $\rho^t_{\mathcal{I}}(\mathcal{A},\mathsf{Uniform})$ is supported on vectors in the Fourier basis of the form $|\hat{f}\rangle$ where*

$$f = y_1\cdot\delta_{x_1}+\cdots+y_s\cdot\delta_{x_s},$$

for some $x_1,\dots,x_s\in X$, $y_1,\dots y_s\in Y$, and $s\in[t]_0$.

In Lemma 7.3.1, we will establish a stronger relationship that directly implies Fact 7.2.6.

We can construct an isometry $\mathsf{Comp}_x : \mathbb{C}[Y]\to\mathbb{C}[Y\cup\{\perp\}]$, for every $x\in X$, that maps the $\mathcal{I}_x$ register to $|\perp\rangle$ if and only if this register contains $|\hat{0}\rangle$, which represents the algorithm knowing nothing about the value stored in register $\mathcal{I}_x$:

$$\mathsf{Comp}_x = |\perp\rangle\langle\hat{0}| + \sum_{z\in Y\setminus\{0\}}|\hat{z}\rangle\langle\hat{z}|.$$

By doing this for every $x\in X$ we obtain the isometry

$$\mathsf{Comp} = \bigotimes_{x\in X}\mathsf{Comp}_x.$$

This isometry Comp *compresses* the information of each of the basis vectors $|\hat{f}\rangle$, for $f = y_1\delta_{x_1}+\cdots+y_s\delta_{x_s}$, in the support of $\rho^t_{\mathcal{I}}(\mathcal{A},\mathsf{Uniform})$, since $\mathsf{Comp}|\hat{f}\rangle\in\mathbb{C}[(Y\cup\{\perp\})^X]$ has $|\perp\rangle$ everywhere except for those $s\leq t$ registers indexed by $x_1,\dots,x_s$. Let us extend QFT_M to $\mathbb{C}[(Y\cup\{\perp\})^X]$ by defining $\mathsf{QFT}_M|\perp\rangle=|\perp\rangle$. We can view

$$|D\rangle = \mathsf{QFT}_M\mathsf{Comp}|\hat{f}\rangle\in\mathbb{C}[(Y\cup\{\perp\})^X]$$

as a *database*, where we have applied QFT_M to bring the databases back to the computational basis. We say that D has size s if $|\{x\in X : D(x)\neq\perp\}|=s$, which we denote by $|D|=s$, and remark that by $\mathsf{QFT}_M|\perp\rangle=|\perp\rangle$, this basis conversion leaves the size of the database unaffected. We write

$$\mathcal{D}_s \coloneqq \{D\in(Y\cup\{\perp\})^X : |D|=s\},\qquad \mathcal{D}_{\leq s}\coloneqq\{D\in(Y\cup\{\perp\})^X : |D|\leq s\}, \tag{7.7}$$

for the sets of all databases of size s and at most s, respectively. In this chapter, we use set notation when working with databases:

- For any $x \in X, y \in Y$ and $D \in (Y \cup \{\perp\})^X$ such that $D(x) = \perp$, we can add a new entry (x, y) to D, to obtain $D' = D \cup (x, y)$. This means that the resulting database D' satisfies $D(x') = D'(x')$ for every $x' \in X \setminus \{x\}$ and $D'(x) = y$.
- For any $x \in X, y \in Y$ and $D \in (Y \cup \{\perp\})^X$ such that $D(x) = y$, we can delete the entry (x, y) from D, to obtain $D' = D \setminus (x, y)$. This means that the resulting database D' satisfies $D(x') = D'(x')$ for every $x' \in X \setminus \{x\}$ and $D'(x) = \perp$.

The compressed oracle gets its name from the fact that each database D of size s can be efficiently represented by the list of pairs $(x_1, D(x_1)), \ldots, (x_s, D(x_s))$, which is bounded in size due to Fact 7.2.6. Hence, the oracle operation $\mathcal{O}_{x,y}$ can be efficiently computed by a quantum algorithm that lazy samples from the uniform distribution, and this circuit (see [CMSZ19] for its explicit construction) is referred to as the compressed (Fourier) oracle:

$$\mathsf{cO}_{x,y} = \mathsf{Comp} \circ \mathcal{O}_{x,y} \circ \mathsf{Comp}^\dagger. \tag{7.8}$$

This framework has many applications in cryptography [CMSZ19, LZ19b, GHHM21, DFMS22] by being able to analyse the interaction of an adversary with a random oracle, which as we have seen is equivalent to where the input register $\mathcal{I}$ is initialised to the uniform superposition over all functions (see (7.4)). In [CMSZ19, HM23], it was shown that this can be generalised to uniform superpositions over distributions where there is no correlation between the values in the registers $\mathcal{I}_x$ and $\mathcal{I}_{x'}$ for distinct $x, x' \in X$. In this chapter, we focus only on the application of the compressed oracle technique to quantum query lower bounds. A rigorous framework of this application has been given in [CFHL21], where the main ingredient of this lower bound (see Theorem 7.2.7 for the full statement) is of the following form:

$$\max_{x \in X, y \in Y} \left\| \mathsf{P}_{\mathcal{D}_\mathcal{R}} \mathsf{cO}_{x,y} \left(I - \mathsf{P}_{\mathcal{D}_\mathcal{R}} \right) \right\|.$$

Here, the relation $\mathcal{R} \subseteq (X \times Y)^k$ defines a set of tuples of size k over $X \times Y$. Each tuple $R \in \mathcal{R}$ is an element of $(X \times Y)^k$ and represents a list of input-output pairs $((x_1, y_1), \ldots, (x_k, y_k))$. Compared to our discussion of search problems in Section 7.2.2, the input $f \in \mathsf{Func}$ is omitted from the relation $\mathcal{R}$. To address this, Theorem 7.2.7 introduces the additional constraint that the input-output pairs in each tuple $R \in \mathcal{R}$ must be consistent with the input f.

The relation $\mathcal{R}$ induces a subset $\mathcal{D}_\mathcal{R} \subseteq \mathcal{D}$, where $D \in \mathcal{D}_\mathcal{R}$ if and only if it is consistent with one of the tuples in $\mathcal{R}$, meaning that there exists a $k \in [N]$ and $R = ((x_1, y_1), \ldots, (x_k, y_k)) \in \mathcal{R}$ such that $D(x_1) = y_1, \ldots, D(x_k) = y_k$. For any subset $A \subseteq \mathcal{D}$, we denote the projection onto this subset as

$$\mathsf{P}_A = \sum_{D \in A} |D\rangle\langle D|. \tag{7.9}$$

Since these projectors project onto computational basis states, we have the added benefit that they commute for distinct choices of A.

Theorem 7.2.7 ([CFHL21])**.** *Fix a finite set X of size N and let $Y = [M-1]_0$. Let $\mathcal{R} \subseteq (X \times Y)^k$ be a relation for some $k \in [M-1]$ and consider a quantum algorithm $\mathcal{A}$ that outputs $(x_1, y_1), \ldots, (x_1, y_k)$. Let p be the probability that both $(x_1, y_1), \ldots, (x_1, y_k) \in \mathcal{R}$ and $y_i = f(x_i)$ for every $i \in [k]$ when $\mathcal{A}$ has interacted with a random oracle, initialised with a uniformly random function f in Y^X. Then:*

$$\sqrt{p} \leq \sum_{t=1}^{T} \max_{x \in X, y \in Y} \left\| \mathsf{P}_{\mathcal{D}_{\leq t} \cap \mathcal{D}_\mathcal{R}} \mathsf{cO}_{x,y} \mathsf{P}_{\mathcal{D}_{\leq t-1} \setminus \mathcal{D}_\mathcal{R}} \right\| + \sqrt{\frac{k}{M}}.$$

Remark 7.2.8. *The framework in [CFHL21] allows for an adversary that makes both sequential as well as parallel queries, whereas we restrict to only the sequential query version of their result. Moreover, they also allow for a series of relations $\mathcal{R}_0, \ldots, \mathcal{R}_T$ instead of a single relation $\mathcal{R}$, where they bound*

$$\left\| \mathsf{P}_{\mathcal{D}_{\leq t} \cap \mathcal{D}_{\mathcal{R}_t}} \mathsf{cO}_{x,y} \mathsf{P}_{\mathcal{D}_{\leq t-1} \setminus \mathcal{D}_{\mathcal{R}_{t-1}}} \right\|.$$

Since the latter generalisation has thus far not been used for any application in the sequential query model, we consider the simplified lower bound as described and applied in [Zha19, LZ19a, HM23].

The form of Theorem 7.2.7 is restricted compared to that of Theorem 7.2.3. We saw that we cannot run $\mathcal{A}$ on any input distribution, but only on Uniform, since Theorem 7.2.7 requires the register $\mathcal{I}$ to be initialised with a uniformly random function f in Y^X. Since $\mathcal{A}$ has to output $(x_1, y_1), \ldots, (x_1, y_k) \in \mathcal{R} \subseteq (X \times Y)^k$, the technique always deals with search problems instead of decision problems. Despite this restriction, it does seem to come with a large advantage compared to the adversary methods. In practice, it appears to be much more straightforward, or at least more intuitive, to come up with a good bound on $\|\mathsf{P}_{\mathcal{D}_{\leq t} \cap \mathcal{D}_{\mathcal{R}}} \mathsf{cO}_{x,y} \mathsf{P}_{\mathcal{D}_{\leq t-1} \setminus \mathcal{D}_{\mathcal{R}}}\|$ than it is to derive a good multiplicative adversary matrix Γ and accompanying constants λ, η, and bound its progress $\|\mathcal{O}_{x,y}^{\dagger} \Gamma^{1/2} \mathcal{O}_{x,y} \Gamma^{-1/2}\|$. Furthermore, it also works well whenever one considers exponentially small success probabilities, whereas the negative-weights adversary method fails in this regime.

7.2.4 Average-case query complexity

We cannot immediately conclude a lower bound on $Q_{1-\epsilon}(\mathcal{F})$ from Theorem 7.2.7. Recall from Definition 7.1.4 and Section 7.2.2 that $Q_{1-\epsilon}(\mathcal{F})$ captures the number of queries required for any quantum query algorithm $\mathcal{A}$ to successfully output $z \in \mathcal{F}(f)$ for any input $f \in \mathsf{Func}$ with success probability at least $1 - \epsilon$. By convexity, $Q_{1-\epsilon}(\mathcal{F})$ is lower bounded by the number of queries required for any input distribution σ, since:

$$\Pr_{f \sim \sigma}[\mathcal{A} \text{ outputs } z \in \mathcal{F}(f)] \geq \min_{f \in \mathsf{Func}} \Pr[\mathcal{A} \text{ outputs } z \in \mathcal{F}(f)].$$

However, $Q_{1-\epsilon}(\mathcal{F})$ is not an interesting metric in the case where $\min_{f \in \mathsf{Func}} \Pr[\mathcal{A} \text{ outputs } z \in \mathcal{F}(f)] = 0$, i.e. when there exists an input $f \in \mathsf{Func}$ when the algorithm *can't* successfully output $z \in \mathcal{F}(f)$ for any input $f \in \mathsf{Func}$. This can occur in Theorem 7.2.7, as the input distribution σ is Uniform. For instance, consider the *collision* problem, where for any input $f \in Y^X$, the goal is to output a pair (x_1, y), (x_2, y) such that $f(x_1) = f(x_2) = y$, referred to as a *collision*. Some inputs $f \in Y^X$ may contain no collisions, making it impossible for the quantum algorithm to output $z \in \mathcal{F}(f)$.

Additionally, even if the worst-case input admits a non-zero probability of success, it can often be more meaningful to show that the problem is hard *on average* rather than merely demonstrating the existence of an input where the problem is hard. This is particularly relevant in the context of cryptography, where it is more desirable to know that most cryptographic keys yield a secure construction than to prove that there exists a single cryptographic key ensuring security.

Therefore, in the remainder of this chapter, we focus on deriving a lower bound for the *average-case* complexity $Q_{1-\epsilon}(\mathcal{F})$ (see, e.g., [AdW01] for further details) rather than the *worst-case* complexity $Q_{1-\epsilon}(\mathcal{F})$:

Definition 7.2.9 (ϵ-error Average-Case Quantum Query Complexity)**.** *Fix $\mathcal{F} : \mathsf{Func} \to 2^{\Sigma}$. Then the* ϵ-error average-case quantum query complexity *of $\mathcal{F}$ and input distribution σ on*

Func*, denoted by $Q^{\sigma}_{1-\epsilon}(\mathcal{F})$, is the minimum number of queries needed by any quantum query algorithm $\mathcal{A}$ such that*

$$\Pr_{f\sim\sigma}[\mathcal{A} \text{ outputs } z \in \mathcal{F}(f)] \geq 1-\epsilon.$$

We can now use Theorem 7.2.7 to lower bound $Q^{\mathsf{Uniform}}_{1-\epsilon}(\mathcal{F})$:

Corollary 7.2.10. *Fix a finite set X of size N and let $Y = [M-1]_0$. Let $\mathcal{R} \subseteq (X\times Y)^k$ be a relation for some $k \in [M-1]$. Then for any $\epsilon \in (0, 1-k/M)$ and any problem $\mathcal{F} : Y^X \to 2^{\mathcal{R}}$, the ϵ-error average-case quantum query complexity $Q^{\mathsf{Uniform}}_{1-\epsilon}(\mathcal{F})$ is lower bounded by the smallest T satisfying*

$$\sqrt{1-\epsilon} - \sqrt{\frac{k}{M}} \leq \sum_{t=1}^{T} \max_{x\in X, y\in Y} \left\| \mathsf{P}_{\mathcal{D}_{\leq t}\cap\mathcal{D}_{\mathcal{R}}} \mathsf{cO}_{x,y} \mathsf{P}_{\mathcal{D}_{\leq t-1}\setminus\mathcal{D}_{\mathcal{R}}} \right\|.$$

Corollary 7.2.10 is slightly less conveniently phrased compared to Corollary 7.2.4 due to its dependence on t in the term

$$\left\| \mathsf{P}_{\mathcal{D}_{\leq t}\cap\mathcal{D}_{\mathcal{R}}} \mathsf{cO}_{x,y} \mathsf{P}_{\mathcal{D}_{\leq t-1}\setminus\mathcal{D}_{\mathcal{R}}} \right\|.$$

As an example of how to determine the "smallest T" in Corollary 7.2.10, we consider the collision problem. In [Zha19], it is shown that for the collision relation

$$\mathcal{R} = \{((x_1,y),(x_2,y)) \in (X\times Y)^2 : f(x_1) = f(x_2) = y\},$$

we can bound

$$\max_{x\in X, y\in Y} \left\| \mathsf{P}_{\mathcal{D}_{\leq t}\cap\mathcal{D}_{\mathcal{R}}} \mathsf{cO}_{x,y} \mathsf{P}_{\mathcal{D}_{\leq t-1}\setminus\mathcal{D}_{\mathcal{R}}} \right\| \leq \sqrt{\frac{t-1}{M}}.$$

Hence, $Q^{\mathsf{Uniform}}_{1-\epsilon}(\mathcal{F})$ is lower bounded by the smallest T satisfying:

$$\sqrt{1-\epsilon} - \sqrt{\frac{2}{M}} \leq \sum_{t=1}^{T} \sqrt{\frac{t-1}{M}} \leq \frac{T^{3/2}}{\sqrt{M}},$$

which can be rearranged to yield

$$T \geq \left(\sqrt{1-\epsilon} - \sqrt{\frac{2}{M}}\right)^{2/3} M^{1/3}.$$

7.3 Multiplicative ladder adversary method

Here, we propose a simplified version of the multiplicative adversary method, that we name the *multiplicative ladder adversary* (MLA) method, which we later prove in Section 7.4 has the compressed oracle technique as a special case. The MLA method is weaker than the multiplicative adversary method as it only considers a subset of all possible multiplicative adversary matrices Γ, which we refer to as MLA matrices, but despite this restriction, it still exhibits a strong direct product theorem, as will be shown in Section 7.5.

7.3.1 Making the adversary matrix time-dependent

Before we define these MLA matrices in Definition 7.3.2, we first provide some motivation behind their definition. In Section 7.2.3, we saw that the compressed oracle seems to make more explicit use of the number of queries to compute the incremental progress by decomposing the set of all possible databases $\mathcal{D} = \bigsqcup_{t=0}^{N} \mathcal{D}_t$ based on their sizes and

integrating these into the projection $\mathsf{P}_{\mathcal{D}_{\mathcal{R}}}$. We generalise this notion by introducing the following construction, that captures the subspace of $\mathbb{C}[Y^X]$ that is reachable from $|\sigma\rangle$ after a fixed number of queries.

First, we define a few components necessary for our construction. Let σ be an initial distribution on $\mathsf{Func} \subseteq Y^X$. For any $t \in [N]$ and any choice of $x_1, \ldots, x_t \in X$ and $y_1, \ldots, y_t \in Y$, define

$$|v^{y_1,\ldots,y_t}_{x_1,\ldots,x_t}\rangle := \frac{1}{\sqrt{\alpha^{y_1,\ldots,y_t}_{x_1,\ldots,x_t}}} \sum_{\substack{f\in\mathsf{Func}:\\ \forall i\in[t], f(x_i)=y_i}} \sqrt{\sigma(f)}|f\rangle, \tag{7.10}$$

where $\alpha^{y_1,\ldots,y_t}_{x_1,\ldots,x_t}$ is the normalisation factor, defined as

$$\alpha^{y_1,\ldots,y_t}_{x_1,\ldots,x_t} := \sum_{\substack{f\in\mathsf{Func}:\\ \forall i\in[t], f(x_i)=y_i}} \sigma(f). \tag{7.11}$$

Lemma 7.3.1. *Define the sequence of subspaces* $\mathsf{Space}_t(\sigma)$ *as follows:*

- *For* $t = 0$*, let* $\mathsf{Space}_0(\sigma) = \mathrm{span}\{|\sigma\rangle\}$*, where* $|\sigma\rangle = \sum_{f\in\mathsf{Func}} \sqrt{\sigma(f)}|f\rangle$ *is the initial state of the input register* $\mathcal{I}$*.*
- *For* $t \in [N]$*, set*

$$\mathsf{Space}_t(\sigma) \coloneqq \mathrm{span}\left\{|v^{y_1,\ldots,y_t}_{x_1,\ldots,x_t}\rangle : (x_i, y_i) \in X \times Y \text{ for } i = 1, \ldots, t\right\}.$$

- *For* $t > N$*, define* $\mathsf{Space}_t(\sigma) = \mathsf{Space}_N(\sigma)$*.*

Then each space $\mathsf{Space}_t(\sigma)$ *represents the subspace of* $\mathbb{C}[Y^X]$ *that is reachable from* $|\sigma\rangle$ *after* t *queries:*

- *For every* $t \in [N]_0$*, there exists a* t*-query quantum algorithm* $\mathcal{A}$*, such that*

$$\mathsf{Space}_t(\sigma) \subseteq \mathrm{span}\left\{\mathrm{supp}\left(\rho^t_{\mathcal{I}}(\mathcal{A}, \sigma)\right)\right\}.$$

- *For every* $t \in [N]_0$ *and* t*-query quantum algorithm* $\mathcal{A}$*, we have*

$$\mathsf{Space}_t(\sigma) \supseteq \mathrm{span}\left\{\mathrm{supp}\left(\rho^t_{\mathcal{I}}(\mathcal{A}, \sigma)\right)\right\}.$$

Before we prove the lemma, we discuss some of its implications. First of all, we find that $\mathsf{Space}_N(\sigma) = \mathrm{span}\{|f\rangle : f \in \mathrm{supp}(\sigma)\}$. Moreover, in the special case where $|\sigma\rangle = |\mathsf{Uniform}\rangle$, we have that

$$\mathsf{Space}_t(\mathsf{Uniform}) = \mathsf{Comp}^\dagger \left(\mathrm{span}\{|D\rangle : D \in \mathcal{D}_{\leq t}\}\right) \mathsf{Comp}, \tag{7.12}$$

which recovers Fact 7.2.6.

We can combine Γ with the projection $\Pi_{\leq t}$ that projects onto $\mathsf{Space}_t(\sigma)$, to ensure that the progress keeps track of the number of queries done by the algorithm. The "$\leq$" in the subscript of each of the projectors $\Pi_{\leq t}$ is there to emphasise that $\Pi_{\leq t-1} \preceq \Pi_{\leq t}$. This is due to the fact that we can let $(x_t, y_t) = (x_{t-1}, y_{t-1})$ in $|v^{y_1,\ldots,y_t}_{x_1,\ldots,x_t}\rangle$. For any T-quantum algorithm $\mathcal{A}$, initial distribution σ, $t \in [T]_0$, and multiplicative adversary matrix Γ, we have

$$W^t(\Gamma, \mathcal{A}) = \mathrm{Tr}\left[\Gamma\rho^t_{\mathcal{I}}(\mathcal{A}, \sigma)\right] = \mathrm{Tr}\left[\Gamma\Pi_{\leq t}\rho^t_{\mathcal{I}}(\mathcal{A}, \sigma)\right] = \mathrm{Tr}\left[\Gamma\rho^t_{\mathcal{I}}(\mathcal{A}, \sigma)\Pi_{\leq t}\right]. \tag{7.13}$$

Proof of Lemma 7.3.1. To prove the first inclusion in Lemma 7.3.1, we show that for any fixed $\boldsymbol{x} = (x_1, \ldots, x_t) \in X^t$ and $\boldsymbol{y} = (y_1, \ldots, y_t) \in Y^t$, we can construct a quantum algorithm A such that

$$|v^{y_1,\ldots,y_t}_{x_1,\ldots,x_t}\rangle \in \operatorname{supp}\left(\rho^t_{\mathcal{I}}(\mathcal{A},\sigma)\right).$$

Let $\mathcal{A}$ be the t-query algorithm that computes $|f(x_1),\ldots,f(x_t)\rangle_{\mathcal{W}}$ in its working register using t queries and $t+1$ unitaries. Additionally, its final unitary U_t uncomputes the $\mathcal{Y}$ register. Then the final state of the algorithm A is

$$|\psi_t(\mathcal{A},\sigma)\rangle = \sum_{f\in\mathsf{Func}} \sqrt{\sigma(f)}|f(x_1),\ldots,f(x_t)\rangle_{\mathcal{W}}|x_t\rangle_{\mathcal{X}}|0\rangle_{\mathcal{Y}}|f\rangle_{\mathcal{I}}.$$

By tracing out all but the input register $\mathcal{I}$, we obtain (see (7.10) and (7.11)):

$$\rho^t_{\mathcal{I}}(\mathcal{A},\sigma) = \sum_{\boldsymbol{y}\in Y^t} \frac{1}{\alpha^{\boldsymbol{y}}_{\boldsymbol{x}}}|v^{\boldsymbol{y}}_{\boldsymbol{x}}\rangle\langle v^{\boldsymbol{y}}_{\boldsymbol{x}}|. \tag{7.14}$$

For the second inclusion in Lemma 7.3.1, we show inductively that for every quantum algorithm $\mathcal{A}$ and $t \in [N]_0$, we have

$$|\psi_t(\mathcal{A},\sigma)\rangle \in \mathcal{WXY}\otimes \mathsf{Space}_t(\sigma).$$

For $t = 0$, we know for any quantum algorithm $\mathcal{A}$ that

$$|\psi_0(\mathcal{A},\sigma)\rangle = U_0|0\rangle_{\mathcal{WXY}}\otimes|\sigma\rangle_{\mathcal{I}} \in \mathcal{WXY}\otimes\mathsf{Space}_t(\sigma),$$

since U_0 acts non-trivially only on the $\mathcal{WXY}$ registers. Now suppose that (7.14) holds for some choice of $t \in [N-1]_0$ and quantum algorithm $\mathcal{A}$, meaning there exist complex coefficients $\beta_{w,x,y,\boldsymbol{x},\boldsymbol{y}}$ satisfying

$$|\psi_t(\mathcal{A},\sigma)\rangle = \sum_{\substack{x\in X, y\in Y, w\in W,\\ \boldsymbol{x}\in X^t, \boldsymbol{y}\in Y}} \beta_{w,x,y,\boldsymbol{x},\boldsymbol{y}}|w,x,\hat{y}\rangle_{\mathcal{WXY}}|v^{\boldsymbol{y}}_{\boldsymbol{x}}\rangle_{\mathcal{I}}.$$

Here $|v^{\boldsymbol{y}}_{\boldsymbol{x}}\rangle_{\mathcal{I}} = |\sigma\rangle$ if $t = 0$. For each $x \in X$, we can decompose the state $|v^{\boldsymbol{y}}_{\boldsymbol{x}}\rangle$ based on the value of $f(x)$ in the computational basis states $|f\rangle$ in $|v^{\boldsymbol{y}}_{\boldsymbol{x}}\rangle$:

$$|\psi_t(\mathcal{A},\sigma)\rangle = \sum_{\substack{x\in X, y\in Y, w\in W,\\ \boldsymbol{x}\in X^t, \boldsymbol{y}\in Y}} \beta_{w,x,y,\boldsymbol{x},\boldsymbol{y}}|w,x,\hat{y}\rangle_{\mathcal{WXY}}\left(\frac{1}{\sqrt{\alpha^{\boldsymbol{y}}_{\boldsymbol{x}}}}\sum_{y_{t+1}\in Y}\sqrt{\alpha^{\boldsymbol{y},y_{t+1}}_{\boldsymbol{x},x}}|v^{\boldsymbol{y},y_{t+1}}_{\boldsymbol{x},x}\rangle\right), \tag{7.15}$$

which is an element of $\mathcal{WXY}\otimes\mathsf{Space}_{t+1}(\sigma)$. Here $\alpha^{\boldsymbol{y}}_{\boldsymbol{x}} = 1$ if $t = 0$. For each such state $|v^{\boldsymbol{y},y_{t+1}}_{\boldsymbol{x},x}\rangle$, we only pick up a global phase when applying a phase query:

$$\mathcal{O}|x,\hat{y}\rangle_{\mathcal{XY}}|v^{y_1,\ldots,y_t,y_{t+1}}_{x_1,\ldots,x_t,x}\rangle = e^{\frac{2\pi i}{M}y\cdot y_{t+1}}|x,\hat{y}\rangle_{\mathcal{XY}}|v^{y_1,\ldots,y_t,y_{t+1}}_{x_1,\ldots,x_t,x}\rangle. \tag{7.16}$$

Moreover, since the unitary U_t acts non-trivially only on the $\mathcal{WXY}$ registers, we find that

$$|\psi_{t+1}(\mathcal{A},\sigma)\rangle = U_t\mathcal{O}|\psi_t(\mathcal{A},\sigma)\rangle \in \mathcal{WXY}\otimes\mathsf{Space}_{t+1}(\sigma). \qquad \square$$

7.3.2 Mapping the progress onto a ladder

The structure of the database projections $\mathsf{P}_{\mathcal{D}_{\leq t}\cap\mathcal{D}_{\mathcal{R}}}$ and $\mathsf{P}_{\mathcal{D}_{\leq t-1}\setminus\mathcal{D}_{\mathcal{R}}}$ in the compressed oracle technique (see Theorem 7.2.7) is, in practice, more convenient to work with than the more abstract projections Λ_i. This is because these projections are built from the database

basis states, which are more intuitive and allow for easy tracking of their sizes with each query (see Fact 7.2.6).

We aim to establish a similar structure on the eigenspaces of Γ. These eigenspaces should resemble steps on a ladder, where each query moves the state up or down by at most one step. Additionally, these steps should be evenly spaced. To formalise this idea, we impose structural constraints on the spectral decomposition of Γ:

$$\Gamma = \sum_{i=0}^{\ell} \lambda_i \Lambda_i. \tag{7.17}$$

Here, $\ell+1$ denotes the number of distinct eigenvalues of Γ, which are sorted in ascending order, and each Λ_i is the projector onto the eigenspace associated with the eigenvalue λ_i.

Definition 7.3.2 (Multiplicative Ladder Adversary Matrix)**.** *Let $\Gamma = \sum_{i=0}^{\ell} \lambda_i \Lambda_i$ be a multiplicative adversary matrix. We say that Γ is a multiplicative ladder adversary (MLA) matrix if the following conditions hold:*

- *The eigenvalues of Γ satisfy $\lambda_i = \kappa^i$ for some $\kappa > 1$, so that*

$$\Gamma = \sum_{i=0}^{\ell} \kappa^i \Lambda_i.$$

- *For every $t \in [N]_0$, Γ commutes with $\Pi_{\leq t}$.*
- *For all $x \in \mathcal{X}$, $y \in \mathcal{Y}$, and $i, i' \in [\ell]_0$, the projections onto the eigenspaces satisfy*

$$\|\Lambda_{i'} \mathcal{O}_{x,y} \Lambda_i\| = 0, \quad \text{if } \left|i' - i\right| > 1. \tag{7.18}$$

The condition expressed in (7.18) ensures that each query can move the state up or down by at most a single eigenspace. Meanwhile, the construction $\Gamma = \sum_{i=0}^{\ell} \kappa^i \Lambda_i$ ensures that the multiplicative progress between successive eigenspaces is constant, specifically a factor of κ.

It might initially seem restrictive to consider only multiplicative adversary matrices that commute with $\Pi_{\leq t}$ for all values of t, given that we are only guaranteed that Γ commutes with $|\sigma\rangle\langle\sigma| = \Pi_{\leq 0}$, since $|\sigma\rangle$ is a 1-eigenvector of Γ. However, the following lemma demonstrates that this is not a restriction for the class of all multiplicative adversary matrices:

Lemma 7.3.3. *Let Γ be a multiplicative adversary matrix with 1-eigenstate $|\sigma\rangle$. Then there exists a multiplicative adversary matrix Γ' with 1-eigenstate $|\sigma\rangle$ such that for every $t \in [N]_0$:*

- *Γ' commutes with $\Pi_{\leq t}$,*
- *$W^t(\Gamma', \mathcal{A}) = W^t(\Gamma, \mathcal{A})$ for every t-query quantum algorithm $\mathcal{A}$.*

Proof. By construction of $\mathsf{Space}_t(\sigma)$, we know that for every $t \in [N]_0$, $\mathsf{Space}_t(\sigma)$ is contained in $\mathsf{Space}_N(\sigma) = \mathrm{span}\{|f\rangle : f \in \mathrm{supp}(\sigma)\}$. Hence, there exists an eigenbasis $\{|e_j\rangle\}_{j \in [|\mathsf{Func}|]}$ for $\mathsf{Space}_N(\sigma)$ that also diagonalises each $\Pi_{\leq t}$:

$$\Pi_{\leq t} = \sum_{j=1}^{|\mathsf{Func}|} \underbrace{\lambda_{t,j}}_{\in\{0,1\}} |e_j\rangle\langle e_j|. \tag{7.19}$$

Since $|\sigma\rangle$ lies in the 1-eigenspace of $\Pi_{\leq 0}$, and therefore also in the 1-eigenspace of every $\Pi_{\leq t}$, we can assume without loss of generality that $|e_1\rangle = |\sigma\rangle$. We construct Γ' using this eigenbasis:

$$\Gamma' := \sum_{j=1}^{|\mathsf{Func}|} \langle e_j|\Gamma|e_j\rangle |e_j\rangle\langle e_j|.$$

We now show that Γ' satisfies the requirements stated in the lemma. First, since $|\sigma\rangle = |e_1\rangle$ is a 1-eigenstate of Γ, we also have

$$\Gamma'|\sigma\rangle = \sum_{j=1}^{|\mathsf{Func}|} \langle e_j|\Gamma|e_j\rangle |e_j\rangle\langle e_j||\sigma\rangle = \langle\sigma|\Gamma|\sigma\rangle|\sigma\rangle = |\sigma\rangle.$$

Moreover, since Γ' is diagonal in the shared eigenbasis $\{|e_j\rangle\}_{j\in[|\mathsf{Func}|]}$ for all $\Pi_{\leq t}$, it commutes with every $\Pi_{\leq t}$. Finally, we find from (7.13) and (7.19) that

$$W^t(\Gamma', \mathcal{A}) = \mathrm{Tr}\left[\Gamma'\Pi_{\leq t}\rho^t_{\mathcal{I}}(\mathcal{A},\sigma)\right] = \mathrm{Tr}\left[\sum_{j=1}^{|\mathsf{Func}|} \lambda_{t,j}\langle e_j|\Gamma|e_j\rangle |e_j\rangle\langle e_j|\rho^t_{\mathcal{I}}(\mathcal{A},\sigma)\right]$$
$$= \mathrm{Tr}\left[\sum_{j=1}^{|\mathsf{Func}|} \lambda^2_{t,j}\langle e_j|\Gamma|e_j\rangle\langle e_j|\rho^t_{\mathcal{I}}(\mathcal{A},\sigma)|e_j\rangle\right] = \mathrm{Tr}\left[\Gamma\Pi_{\leq t}\rho^t_{\mathcal{I}}(\mathcal{A},\sigma)\Pi_{\leq t}\right] = W^t(\Gamma,\mathcal{A}).$$

□

This new definition allows us to prove an MLA-version of Theorem 7.2.3. This result is strictly weaker, as it only considers a subset of all possible multiplicative adversary matrices, but it greatly simplifies the upper bound on the progress achievable in a single query (Item 2).

Theorem 7.3.4. *Fix a problem $\mathcal{F} : \mathsf{Func} \to 2^\Sigma$, an input distribution σ on Func, a constant $\kappa > 1$, and an MLA matrix $\Gamma = \sum_{i=0}^{\ell} \kappa^i \Lambda_i$ with 1-eigenstate $|\sigma\rangle$ (see Definition 7.3.2). Let λ be a real number with $1 < \lambda \leq \kappa^\ell$. Let Λ_{bad} be the projector onto the eigenspaces of Γ corresponding to eigenvalues smaller than λ and let $\eta \leq 1-\epsilon$ be a positive constant such that $\|F_z\Lambda_{\mathsf{bad}}\|^2 \leq \eta$ for every $z \in \Sigma$, where $F_z = \sum_{\substack{f\in\mathsf{Func}:\\ \mathcal{F}(f)\ni z}} |f\rangle\langle f|$. Then:*

1. *For any quantum algorithm $\mathcal{A}$, $W^0(\Gamma,\mathcal{A}) = 1$.*
2. *For any T-query quantum algorithm $\mathcal{A}$, and $t \in [T-1]_0$,*

$$\frac{W^{t+1}(\Gamma,\mathcal{A})}{W^t(\Gamma,\mathcal{A})} \leq \max_{\substack{i\in[\ell-1]_0,\\ x\in X, y\in Y}} \left(1 + \max_{\substack{i\in[\ell-1]_0,\\ x\in X, y\in Y}} \frac{\kappa-1}{\sqrt{\kappa}} \|\Lambda_{i+1}\Pi_{\leq t+1}\mathcal{O}_{x,y}\Pi_{\leq t}\Lambda_i\|\right)^2$$

3. *For any T-query quantum algorithm $\mathcal{A}$ that solves $\mathcal{F}$ on input $|\sigma\rangle$ with success probability at least $1-\epsilon$, $W^T(\Gamma,\mathcal{A}) \geq 1 + (\lambda-1)(\sqrt{1-\epsilon} - \sqrt{\eta})^2$.*

Note that the upper bound on W^{t+1}/W^t, the progress made in one step now depends on t. This is necessary to capture the power of the compressed oracle method, where, for example, the probability of (i.e. amplitude on) finding a collision in a single query is greater the more queried values you have stored in memory.

Corollary 7.3.5. *For any η that satisfies the constraints of Theorem 7.3.4, any $\epsilon \in (0, 1-\eta)$, problem $\mathcal{F} : \mathsf{Func} \to 2^{\Sigma}$, and input distribution σ on Func, the ϵ-error average-case quantum query complexity $Q^{\sigma}_{1-\epsilon}(\mathcal{F})$ is lower bounded by the smallest T such that satisfying*

$$1 \leq \min_{\Gamma,\lambda} \left(-(\lambda-1)(\sqrt{1-\epsilon}-\sqrt{\eta})^2 + \prod_{t=1}^{T} \left(1 + \max_{\substack{i\in[\ell-1]_0,\\ x\in X, y\in Y}} \frac{\kappa-1}{\sqrt{\kappa}} \left\| \Lambda_{i+1} \Pi_{\leq t} \mathcal{O}_{x,y} \Pi_{\leq t-1} \Lambda_i \right\| \right)^2 \right),$$

Proof of Theorem 7.3.4. Item 1 and 3 follow from Theorem 7.2.3 and Lemma 7.2.5, so we focus on proving Item 2. We fix any T-query algorithm $\mathcal{A}$ and begin by following similar steps as in [Špa08] to upper bound the ratio

$$\frac{W^{t+1}(\Gamma, \mathcal{A})}{W^{t}(\Gamma, \mathcal{A})}.$$

Observe that the query operator $\mathcal{O}$ cannot directly be inserted into the progress measure since it acts on all registers $\mathcal{XYI}$, whereas each MLA matrix is only defined on $\mathcal{I}$. Thus, we lift Γ to this larger space by constructing $\Upsilon = I_{\mathcal{WXY}} \otimes \Gamma$, which immediately yields

$$W^{t}(\Gamma, \mathcal{A}) = \mathrm{Tr}\left[\Upsilon |\psi_t(\mathcal{A}, \sigma)\rangle\langle\psi_t(\mathcal{A}, \sigma)|\right].$$

Because $\mathcal{A}$ and $|\sigma\rangle$ are fixed, we simplify the notation in the rest of the proof and omit $(\mathcal{A}, \sigma)$. The addition of Υ results in

$$\begin{aligned}\frac{W^{t+1}(\Gamma, \mathcal{A})}{W^{t}(\Gamma, \mathcal{A})} &= \frac{\mathrm{Tr}\left[\Upsilon |\psi_{t+1}\rangle\langle\psi_{t+1}|\right]}{\mathrm{Tr}\left[\Upsilon |\psi_t\rangle\langle\psi_t|\right]} = \frac{\mathrm{Tr}\left[\Upsilon U_{t+1}\mathcal{O}|\psi_t\rangle\langle\psi_t|\mathcal{O}^{\dagger}U_{t+1}^{\dagger}\right]}{\mathrm{Tr}\left[\Upsilon |\psi_t\rangle\langle\psi_t|\right]} \\ &= \frac{\mathrm{Tr}\left[\mathcal{O}^{\dagger}U_{t+1}^{\dagger}\Upsilon U_{t+1}\mathcal{O}|\psi_t\rangle\langle\psi_t|\right]}{\mathrm{Tr}\left[\Upsilon |\psi_t\rangle\langle\psi_t|\right]},\end{aligned}$$

where in the final equality we have used the cyclic property of the trace. Since the unitary U_{t+1} acts as the identity on register $\mathcal{I}$, we obtain that $U_{t+1}^{\dagger}\Upsilon U_{t+1} = \Upsilon$. This allows us to simplify

$$\frac{\mathrm{Tr}\left[\mathcal{O}^{\dagger}U_{t+1}^{\dagger}\Upsilon U_{t+1}\mathcal{O}|\psi_t\rangle\langle\psi_t|\right]}{\mathrm{Tr}\left[\Upsilon |\psi_t\rangle\langle\psi_t|\right]} = \frac{\mathrm{Tr}\left[\mathcal{O}^{\dagger}\Upsilon\mathcal{O}|\psi_t\rangle\langle\psi_t|\right]}{\mathrm{Tr}\left[\Upsilon |\psi_t\rangle\langle\psi_t|\right]}.$$

At this point, we deviate from [Špa08] by making use of the projection $\Pi_{\leq t}$ onto $\mathsf{Space}_t(\sigma)$ from Lemma 7.3.1. Our goal is to show that the following equation holds:

$$\frac{W^{t+1}(\Gamma, \mathcal{A})}{W^{t}(\Gamma, \mathcal{A})} \leq \max_{x\in X, y\in Y} \left\| \Gamma^{1/2}\mathcal{O}_{x,y}\Pi_{\leq t}\Gamma^{-1/2} \right\|^2. \tag{7.20}$$

Let $|\tau\rangle = \Upsilon^{1/2}|\psi_t\rangle$, meaning $|\psi_t\rangle = \Upsilon^{-1/2}|\tau\rangle$. Then

$$\begin{aligned}&\frac{\mathrm{Tr}\left[\mathcal{O}^{\dagger}\Upsilon\mathcal{O}|\psi_t\rangle\langle\psi_t|\right]}{\mathrm{Tr}\left[\Upsilon |\psi_t\rangle\langle\psi_t|\right]} = \frac{\langle\psi_t|\mathcal{O}^{\dagger}\Upsilon\mathcal{O}|\psi_t\rangle}{\langle\psi_t|\Upsilon|\psi_t\rangle} = \frac{\langle\psi_t|\left(I_{\mathcal{WXY}}\otimes\Pi_{\leq t}\right)\mathcal{O}^{\dagger}\Upsilon\mathcal{O}\left(I_{\mathcal{WXY}}\otimes\Pi_{\leq t}\right)|\psi_t\rangle}{\langle\psi_t|\Upsilon|\psi_t\rangle} \\ &= \frac{\langle\tau|\Upsilon^{-1/2}\left(I_{\mathcal{WXY}}\otimes\Pi_{\leq t}\right)\mathcal{O}^{\dagger}\Upsilon\mathcal{O}\left(I_{\mathcal{WXY}}\otimes\Pi_{\leq t}\right)\Upsilon^{-1/2}|\tau\rangle}{\langle\tau|\tau\rangle} \\ &\leq \left\|\Upsilon^{1/2}\mathcal{O}\left(I_{\mathcal{WXY}}\otimes\Pi_{\leq t}\right)\Upsilon^{-1/2}\right\|^2 = \max_{x\in X, y\in Y}\left\|\Gamma^{1/2}\mathcal{O}_{x,y}\Pi_{\leq t}\Gamma^{-1/2}\right\|^2,\end{aligned}$$

where we use the fact that Γ, and hence also Υ, is Hermitian, as well as (7.1). Using the triangle inequality, we can further bound this expression (for any fixed x, y and without the square) as

$$\left\|\Gamma^{1/2}\mathcal{O}_{x,y}\Pi_{\leq t}\Gamma^{-1/2}\right\| \leq \frac{1}{\sqrt{\kappa}} + \left\|\left(\Gamma^{1/2}\mathcal{O}_{x,y}\Pi_{\leq t} - \frac{1}{\sqrt{\kappa}}\mathcal{O}_{x,y}\Pi_{\leq t}\Gamma^{1/2}\right)\Gamma^{-1/2}\right\|. \tag{7.21}$$

Here is where we make the second deviation from [Špa08]. Since the projections onto the eigenspaces of a Hermitian matrix form a resolution of the identity, we can write the matrix

$$\left(\Gamma^{1/2}\mathcal{O}_{x,y}\Pi_{\leq t} - \frac{1}{\sqrt{\kappa}}\mathcal{O}_{x,y}\Pi_{\leq t}\Gamma^{1/2}\right)\Gamma^{-1/2}$$

as a block matrix with entries indexed by $i, i' \in [\ell]_0$, equal to

$$\begin{aligned}
&\Lambda_{i'}\left(\Gamma^{1/2}\mathcal{O}_{x,y}\Pi_{\leq t} - \frac{1}{\sqrt{\kappa}}\mathcal{O}_{x,y}\Pi_{\leq t}\Gamma^{1/2}\right)\Gamma^{-1/2}\Lambda_i \\
&= \left(\sqrt{\kappa^{i'}}\Lambda_{i'}\mathcal{O}_{x,y}\Pi_{\leq t} - \frac{1}{\sqrt{\kappa}}\Lambda_{i'}\mathcal{O}_{x,y}\Pi_{\leq t}\Gamma^{1/2}\right)\Gamma^{-1/2}\Lambda_i \qquad (7.22)\\
&= \frac{\sqrt{\kappa^{i'}}}{\sqrt{\kappa^{i}}}\Lambda_{i'}\mathcal{O}_{x,y}\Pi_{\leq t}\Lambda_i - \frac{1}{\sqrt{\kappa^{1}}}\Lambda_{i'}\mathcal{O}_{x,y}\Lambda_i = \frac{\sqrt{\kappa^{i'-i+1}}-1}{\sqrt{\kappa}}\Lambda_{i'}\mathcal{O}_{x,y}\Pi_{\leq t}\Lambda_i,
\end{aligned}$$

where we used $\Gamma = \sum_{i=0}^{\ell}\kappa^i\Lambda_i$ (see (7.17)) and consequently $\Gamma^{-1} = \sum_{i=0}^{\ell}\kappa^{-i}\Lambda_i$. As Γ is an MLA matrix, all entries in this block matrix must be zero by (7.18), apart from the entries on diagonal, superdiagonal and subdiagonal. The entries on the superdiagonal however are also zero, which can be verified by substituting $i' = i - 1$ in (7.22). This enables us to bound the norm of this block matrix by the block matrix $M_{x,y}$, which will contain only zero blocks, except for the blocks on the diagonal and subdiagonal, which have respective entries a and b (that depend on x and y) multiplied by identity matrices of the appropriate dimensions. We set these values to

$$\begin{aligned}
a &\coloneqq \max_{i\in[\ell-1]_0}\left\|\frac{\sqrt{\kappa^{i-i+1}}-1}{\sqrt{\kappa}}\Lambda_i\mathcal{O}_{x,y}\Pi_{\leq t}\Lambda_i\right\| \leq 1 - \frac{1}{\sqrt{\kappa}},\\
b &\coloneqq \max_{i\in[\ell-1]_0}\left\|\frac{\sqrt{\kappa^{(i+1)-i+1}}-1}{\sqrt{\kappa}}\Lambda_{i+1}\mathcal{O}_{x,y}\Pi_{\leq t}\Lambda_i\right\| = \max_{i\in[\ell-1]_0}\frac{\kappa-1}{\sqrt{\kappa}}\left\|\Lambda_{i+1}\mathcal{O}_{x,y}\Pi_{\leq t}\Lambda_i\right\|.
\end{aligned}$$

Using this new matrix $M_{x,y}$ we can therefore bound

$$\max_{x\in X,y\in Y}\left\|\left(\Gamma^{1/2}\mathcal{O}_{x,y}\Pi_{\leq t} - \frac{1}{\sqrt{\kappa}}\mathcal{O}_{x,y}\Pi_{\leq t}\Gamma^{1/2}\right)\Gamma^{-1/2}\right\| \leq \max_{x\in X,y\in Y}\|M_{x,y}\|. \qquad (7.23)$$

For a block matrix of the form

$$M_{x,y} = \begin{bmatrix} a & 0 & 0 & & \\ b & a & 0 & & \\ 0 & b & \ddots & \ddots & \\ & & \ddots & \ddots & 0 \\ & & & b & a \end{bmatrix},$$

we upper bound its spectral norm by $a + b$, since $\|M_{x,y}\| \leq \sqrt{\|M_{x,y}\|_1\|M_{x,y}\|_\infty}$ (see Lemma 2.1.1). We can now nearly conclude the proof by combining (7.20) with (7.21) and (7.23):

$$\frac{W^{t+1}(\Gamma,\mathcal{A})}{W^t(\Gamma,\mathcal{A})} \leq \left(\frac{1}{\sqrt{\kappa}} + \max_{x\in X,y\in Y}\|M_{x,y}\|\right)^2 \leq \left(1 + \max_{\substack{i\in[\ell-1]_0,\\ x\in X,y\in Y}}\frac{\kappa-1}{\sqrt{\kappa}}\left\|\Lambda_{i+1}\mathcal{O}_{x,y}\Pi_{\leq t}\Lambda_i\right\|\right)^2.$$

To conclude Item 2, we still need to replace $\Lambda_{i+1}\mathcal{O}_{x,y}\Pi_{\leq t}\Lambda_i$ with $\Lambda_{i+1}\Pi_{\leq t+1}\mathcal{O}_{x,y}\Pi_{\leq t}\Lambda_i$. This follows directly from (7.15) and (7.16), which shows that for every one of the basis states $|v^{y_1,\dots,y_t}_{x_1,\dots,x_t}\rangle$ spanning $\mathsf{Space}_t(\sigma)$ and for every $x \in X$ and $y \in Y$, we have:

$$\begin{aligned}\mathcal{O}_{x,y}|v^{y_1,\dots,y_t}_{x_1,\dots,x_t}\rangle &= \frac{1}{\sqrt{\alpha^{y_1,\dots,y_t}_{x_1,\dots,x_t}}} \sum_{y_{t+1}\in Y} \sqrt{\alpha^{y_1,\dots,y_t,y_{t+1}}_{x_1,\dots,x_t,x}}\mathcal{O}_{x,y}|v^{y_1,\dots,y_t,y_{t+1}}_{x_1,\dots,x_t,x}\rangle \\ &= \sum_{y_{t+1}\in Y} \sqrt{\alpha^{y_1,\dots,y_t,y_{t+1}}_{x_1,\dots,x_t,x}} e^{\frac{2\pi\iota}{M}y\cdot y_{t+1}}|v^{y_1,\dots,y_t,y_{t+1}}_{x_1,\dots,x_t,x}\rangle \in \mathsf{Space}_{t+1}(\sigma),\end{aligned}$$

meaning

$$\mathcal{O}_{x,y}\Pi_{\leq t} = \Pi_{\leq t+1}\mathcal{O}_{x,y}\Pi_{\leq t}. \tag{7.24}$$

□

The machinery of MLA matrices is not necessary for the reduction in Section 7.4. For this reduction, we construct multiplicative matrices with $\ell = 1$, which automatically satisfy (7.18). However, a general ℓ is required if we aim to compute a function $\mathcal{F}$ on ℓ independent instances simultaneously, as discussed in Section 7.5. Furthermore, almost all multiplicative adversary matrices constructed so far to establish lower bounds (see [AŠdW06, Špa08, AMRR11]) are, in fact, MLA matrices. This observation suggests that MLA matrices form a natural subset worthy of deeper analysis.

The following is a useful property that follows from (7.24), which we will employ in the subsequent sections:

Fact 7.3.6. *$\|\Lambda_i\Pi_{\leq t}\mathcal{O}_{x,y}\Pi_{\leq t-1}\Lambda_{i-1}\|$ is monotonically non-decreasing in $t \in [N]$ for all $i \in [\ell]$.*

Proof. Let $\Pi_t := \Pi_{\leq t} - \Pi_{\leq t-1}$ be the projection onto $\mathsf{Space}_t(\sigma) \cap \mathsf{Space}_{t-1}(\sigma)^\perp$. Since unitaries preserve inner products, we have by (7.24) that

$$\Pi_{\leq t+1}\mathcal{O}_{x,y}\Pi_t = \mathcal{O}_{x,y}\Pi_t \perp \mathcal{O}_{x,y}\Pi_{\leq t-1} = \Pi_{\leq t}\mathcal{O}_{x,y}\Pi_{\leq t-1}.$$

This means that $\Pi_{\leq t+1}\mathcal{O}_{x,y}\Pi_t$ and $\Pi_{\leq t}\mathcal{O}_{x,y}\Pi_{\leq t-1}$ have orthogonal images and coimages. This orthogonality is preserved after multiplying with Λ_i and Λ_{i-1} since Γ commutes with $\Pi_{\leq t-1}, \Pi_{\leq t}$ and $\Pi_{\leq t+1}$. Hence, we have

$$\begin{aligned}&\|\Lambda_i\Pi_{\leq t+1}\mathcal{O}_{x,y}\Pi_{\leq t}\Lambda_{i-1}\| = \|\Lambda_i\left(\Pi_{\leq t} + \Pi_{t+1}\right)\mathcal{O}_{x,y}\left(\Pi_{\leq t-1} + \Pi_t\right)\Lambda_{i-1}\| \\ &= \|\Lambda_i\left(\Pi_{\leq t}\mathcal{O}_{x,y}\Pi_{\leq t-1} + \Pi_{\leq t}\mathcal{O}_{x,y}\Pi_t + \Pi_{t+1}\mathcal{O}_{x,y}\Pi_{\leq t-1} + \Pi_{t+1}\mathcal{O}_{x,y}\Pi_t\right)\Lambda_{i-1}\| \\ &= \|\Lambda_i\left(\Pi_{\leq t}\mathcal{O}_{x,y}\Pi_{\leq t-1} + \Pi_{\leq t+1}\mathcal{O}_{x,y}\Pi_t\right)\Lambda_{i-1}\| \geq \|\Lambda_i\left(\Pi_{\leq t}\mathcal{O}_{x,y}\Pi_{\leq t-1}\right)\Lambda_{i-1}\|.\end{aligned}$$

□

7.4 The reduction

In this section, we present an explicit reduction from the compressed oracle technique to our new MLA method:

Theorem 7.4.1. *Fix a finite set X of size N and let $Y = [M-1]_0$. Consider a relation $\mathcal{R} \subseteq (X \times Y)^k$ for some $k \in [M-1]$. Let $\epsilon \in \left(0, 1 - (9-4\sqrt{2})\frac{k}{M}\right)$, and fix any problem $\mathcal{F} : Y^X \to 2^{\mathcal{R}}$. Define the quantities $\mathsf{MLADV}^{\mathsf{Uniform}}_{1-\epsilon,\frac{2k}{M}}(\mathcal{F})$ and $\mathsf{COMP}^{\mathsf{Uniform}}_{1-\epsilon}(\mathcal{F})$ as the lower bounds on $Q^{\mathsf{Uniform}}_{1-\epsilon}(\mathcal{F})$ obtained by Corollary 7.3.5 (with η set to $\frac{2k}{M}$) and Corollary 7.2.10, respectively. Then, we have*

$$\mathsf{COMP}^{\mathsf{Uniform}}_{1-\epsilon}(\mathcal{F}) \leq 6 \cdot \mathsf{MLADV}^{\mathsf{Uniform}}_{1-\epsilon,\frac{2k}{M}}(\mathcal{F}).$$

Recall from Corollary 7.2.10 that $\mathsf{COMP}^{\mathsf{Uniform}}_{1-\epsilon}(\mathcal{F})$ is equal to the smallest T satisfying

$$\sqrt{1-\epsilon} - \sqrt{\frac{k}{M}} \leq \sum_{t=1}^{T} \max_{x\in X, y\in Y} \left\| \mathsf{P}_{\mathcal{D}_{\leq t}\cap\mathcal{D}_{\mathcal{R}}} \mathsf{cO}_{x,y} \mathsf{P}_{\mathcal{D}_{\leq t-1}\setminus\mathcal{D}_{\mathcal{R}}} \right\|. \tag{7.25}$$

We start by removing the compressed oracle $\mathsf{cO}_{x,y}$ in (7.25). For $t \in [T]_0$, consider the following projections:

$$\begin{aligned} \Pi_{1,t} &\coloneqq \mathsf{Comp}^\dagger \mathsf{P}_{\mathcal{D}_{\leq t}\cap\mathcal{D}_{\mathcal{R}}} \mathsf{Comp}, \\ \Pi_{0,t} &\coloneqq \mathsf{Comp}^\dagger \mathsf{P}_{\mathcal{D}_{\leq t}\setminus\mathcal{D}_{\mathcal{R}}} \mathsf{Comp}. \end{aligned} \tag{7.26}$$

By the definition of $\mathsf{cO}_{x,y}$ from (7.8), we find that the projections in (7.26) allow us to rewrite the right-hand side of (7.25) as:

$$\begin{aligned} &\sum_{t=1}^{T} \max_{x\in X, y\in Y} \left\| \mathsf{P}_{\mathcal{D}_{\leq t}\cap\mathcal{D}_{\mathcal{R}}} \mathsf{cO}_{x,y} \mathsf{P}_{\mathcal{D}_{\leq t-1}\setminus\mathcal{D}_{\mathcal{R}}} \right\| \\ &= \sum_{t=1}^{T} \max_{x\in X, y\in Y} \left\| \left(\mathsf{Comp}\Pi_{1,t}\mathsf{Comp}^\dagger\right) \left(\mathsf{Comp}\mathcal{O}_{x,y}\mathsf{Comp}^\dagger\right) \left(\mathsf{Comp}\Pi_{0,t-1}\mathsf{Comp}^\dagger\right) \right\| \\ &= \sum_{t=1}^{T} \max_{x\in X, y\in Y} \left\| \Pi_{1,t}\mathcal{O}_{x,y}\Pi_{0,t-1} \right\|. \end{aligned}$$

Hence, by Corollary 7.2.10, $\mathsf{COMP}^{\mathsf{Uniform}}_{1-\epsilon}(\mathcal{F})$ is upper bounded by the smallest value of T satisfying

$$\sqrt{1-\epsilon} - \sqrt{\frac{k}{M}} \leq \sum_{t=1}^{T} \max_{x\in X, y\in Y} \left\| \Pi_{1,t}\mathcal{O}_{x,y}\Pi_{0,t-1} \right\|. \tag{7.27}$$

Next, we show that for any $\mathcal{R} \subseteq (X \times Y)^k$, we can always construct an explicit MLA Γ (see Definition 7.3.2), with accompanying parameter λ and $\eta = \frac{2k}{M}$ satisfying the conditions of Theorem 7.3.4, such that any T that satisfies

$$1 + (\lambda - 1)(\sqrt{1-\epsilon} - \sqrt{\eta})^2 \leq \prod_{t=1}^{T} \max_{\substack{i\in[\ell-1]_0 \\ x\in X, y\in Y}} \left(1 + \frac{\kappa - 1}{\sqrt{\kappa}} \left\| \Lambda_{i+1}\Pi_{\leq t}\mathcal{O}_{x,y}\Pi_{\leq t-1}\Lambda_i \right\| \right)^2$$

also satisfies

$$\sqrt{1-\epsilon} - \sqrt{\frac{k}{M}} \leq \sum_{t=1}^{6T} \max_{x\in X, y\in Y} \left\| \Pi_{1,t}\mathcal{O}_{x,y}\Pi_{0,t-1} \right\|. \tag{7.28}$$

This then proves Theorem 7.4.1 by Corollary 7.3.5 and (7.27).

For $\ell = 1$, we know from Definition 7.3.2 that any multiplicative ladder adversary matrix has the following form for some $\kappa > 1$:

$$\Gamma = \Lambda_0 + \kappa\Lambda_1.$$

We set the eigenspaces of Γ to correspond to the projections $\Lambda_1 \coloneqq \Pi_{1,N}$ (see (7.26)) and $\Lambda_0 \coloneqq I - \Lambda_1$.

Claim 7.4.2. *For each $t \in [T]_0$*

$$\Pi_{\leq t}\Lambda_0 = \Pi_{0,t}, \qquad \Lambda_1\Pi_{\leq t} = \Pi_{1,t}. \tag{7.29}$$

Proof. Since $\mathsf{P}_{\mathcal{D}_{\leq N}}$ is the identity on $\mathbb{C}[(Y \cup \{\perp\})^X]$, we find that

$$\begin{aligned}\Lambda_0 = I - \Lambda_1 &= \mathsf{Comp}^\dagger \mathsf{P}_{\mathcal{D}_{\leq N}} \mathsf{Comp} - \mathsf{Comp}^\dagger \mathsf{P}_{\mathcal{D}_{\leq N} \cap \mathcal{D}_{\mathcal{R}}} \mathsf{Comp} \\ &= \mathsf{Comp}^\dagger \mathsf{P}_{\mathcal{D}_{\leq N} \setminus \mathcal{D}_{\mathcal{R}}} \mathsf{Comp}.\end{aligned}$$

By (7.12) we know that $\Pi_{\leq t} = \mathsf{Comp}^\dagger \mathsf{P}_{\mathcal{D}_{\leq t}} \mathsf{Comp}$. Together with the commutativity of the projectors onto subsets of $\mathcal{D}$ (see (7.9)), this implies that

$$\begin{aligned}\Pi_{\leq t}\Lambda_0 &= \mathsf{Comp}^\dagger \mathsf{P}_{\mathcal{D}_{\leq t}} \mathsf{Comp}\mathsf{Comp}^\dagger \mathsf{P}_{\mathcal{D}_{\leq N} \setminus \mathcal{D}_{\mathcal{R}}} \mathsf{Comp} \\ &= \mathsf{Comp}^\dagger \mathsf{P}_{\mathcal{D}_{\leq t} \setminus \mathcal{D}_{\mathcal{R}}} \mathsf{Comp} = \Pi_{0,t}.\end{aligned}$$

Similarly we have

$$\begin{aligned}\Lambda_1\Pi_{\leq t} &= \mathsf{Comp}^\dagger \mathsf{P}_{\mathcal{D}_{\leq N} \cap \mathcal{D}_{\mathcal{R}}} \mathsf{Comp}\mathsf{Comp}^\dagger \mathsf{P}_{\mathcal{D}_{\leq t}} \mathsf{Comp} \\ &= \mathsf{Comp}^\dagger \mathsf{P}_{\mathcal{D}_{\leq t} \cap \mathcal{D}_{\mathcal{R}}} \mathsf{Comp} = \Pi_{1,t}.\end{aligned}$$

□

Claim 7.4.3. *Let $\Lambda_1 \coloneqq \Pi_{1,N}$ (see (7.26)), $\Lambda_0 = I - \Lambda_1$ and $\Gamma = \Lambda_0 + \kappa\Lambda_1$ for some constant $\kappa > 1$. Then Γ is an MLA matrix as defined in Definition 7.3.2 with $|\mathsf{Uniform}\rangle$ as a 1-eigenvector.*

Proof. It is clear that this construction makes Γ positive definite with smallest eigenvalue 1 and largest eigenvalue κ. Moreover, since $\ell = 1$, we automatically satisfy (7.18) from Definition 7.3.2. Γ also commutes with every $\Pi_{\leq t}$ due to the commutativity of the projectors onto subsets of $\mathcal{D}$. Hence, it only rests us to verify that $|\mathsf{Uniform}\rangle = \frac{1}{\sqrt{M^N}} \sum_{f \in Y^X} |f\rangle$ is indeed an eigenvector of Γ with eigenvalue 1:

$$\Lambda_0|\mathsf{Uniform}\rangle = |\mathsf{Uniform}\rangle - \Lambda_1|\mathsf{Uniform}\rangle = |\mathsf{Uniform}\rangle - \mathsf{Comp}^\dagger \mathsf{P}_{\mathcal{D}_{\mathcal{R}}} |\perp\rangle^{\otimes N} = |\mathsf{Uniform}\rangle,$$

since the empty database $|\perp\rangle^{\otimes N}$ can never be an element of $\mathcal{D}_{\mathcal{R}}$. □

Knowing that Γ is an MLA with $|\mathsf{Uniform}\rangle$ as a 1-eigenvector, we may apply Corollary 7.3.5. By taking the natural logarithm of both sides, it states

$$\ln\left(1 + (\lambda - 1)(\sqrt{1-\epsilon} - \sqrt{\eta})^2\right) \leq 2 \sum_{t=1}^{T} \ln\left(1 + \max_{x \in X, y \in Y} \frac{\kappa - 1}{\sqrt{\kappa}} \left\|\Lambda_1 \Pi_{\leq t} \mathcal{O}_{x,y} \Pi_{\leq t-1} \Lambda_0\right\|\right).$$

To show that this implies (7.27), we set $\lambda = \kappa = 1 + (e-1)/\left(\sqrt{1-\epsilon} - \sqrt{\eta}\right)^2$ and multiply both sides of the equation with $\sqrt{1-\epsilon} - \sqrt{\eta}$ to arrive at

$$\begin{aligned}\sqrt{1-\epsilon} - \sqrt{\eta} &\leq 2\left(\sqrt{1-\epsilon} - \sqrt{\eta}\right) \sum_{t=1}^{T} \ln\left(1 + \max_{x \in X, y \in Y} \frac{\kappa - 1}{\sqrt{\kappa}} \left\|\Lambda_1 \Pi_{\leq t} \mathcal{O}_{x,y} \Pi_{\leq t-1} \Lambda_0\right\|\right) \\ &\leq 2\left(\sqrt{1-\epsilon} - \sqrt{\eta}\right) \frac{\kappa - 1}{\sqrt{\kappa}} \sum_{t=1}^{T} \max_{x \in X, y \in Y} \left\|\Lambda_1 \Pi_{\leq t} \mathcal{O}_{x,y} \Pi_{\leq t-1} \Lambda_0\right\| \\ &\leq 3 \sum_{t=1}^{T} \max_{x \in X, y \in Y} \left\|\Lambda_1 \Pi_{\leq t} \mathcal{O}_{x,y} \Pi_{\leq t-1} \Lambda_0\right\|. \qquad (7.30)\end{aligned}$$

To finalise the proof, we show that the choice of $\eta = \frac{2k}{M}$ satisfies the conditions of Theorem 7.3.4. By our choice of Γ, λ, κ, the projection Λ_{bad} is equal to Λ_0. The proof of the following lemma can be skipped if the reader is familiar with the compressed oracle technique, as the technique is reminiscent to the proof of the lemma in [Zha19] that links the compressed Fourier oracle to the original oracle.

Lemma 7.4.4. *Let $\Gamma = \Lambda_0 + \kappa\Lambda_1$ be a multiplicative adversary matrix (see Definition 7.2.1) with $\Lambda_1 = \mathsf{Comp}^\dagger \mathsf{P}_{\mathcal{D}_\mathcal{R}} \mathsf{Comp}$ and $\Lambda_0 = I - \Lambda_1$. Then for every $z \in \mathcal{R} \subseteq (X \times Y)^k$ we have*

$$\|F_z \Lambda_0\|^2 \leq \frac{2k}{M},$$

where $F_z = \sum_{\substack{f \in Y^X: \\ \mathcal{F}(f) \ni z}} |f\rangle\langle f|$.

Proof. We know that z is of the form $(x_1, y_1), \ldots, (x_k, y_k)$. Hence, we have that the projector F_z is equal to $F_z = \bigotimes_{i=1}^k |y_i\rangle\langle y_i|_{\mathcal{I}_{x_i}}$ (and acts as the identity on all other registers of $\mathcal{I}$). F_z is therefore equal to $\mathsf{P}_{\mathcal{D}_{\{z\}}}$, but the latter acts on $\mathbb{C}[(Y \cup \{\perp\})^X]$, whereas the former acts on $\mathbb{C}[Y^X]$. By definition of Λ_0, we find that

$$\Lambda_0 = I - \mathsf{Comp}^\dagger \mathsf{P}_\mathcal{R} \mathsf{Comp} \preceq I - \mathsf{Comp}^\dagger \mathsf{P}_{\mathcal{D}_{\{z\}}} \mathsf{Comp}.$$

Combined with the projection F_z this yields

$$\|F_z \Lambda_0\| \leq \left\| F_z - F_z \mathsf{Comp}^\dagger \mathsf{P}_{\mathcal{D}_{\{z\}}} \mathsf{Comp} \right\|. \tag{7.31}$$

The projections F_z and $\mathsf{P}_{\mathcal{D}_{\{z\}}}$ are easier to analyse if we view each $\mathcal{I}_{x_i}$ register in the Fourier basis. If we abuse the equality sign, since both projectors act on slightly different Hilbert spaces, they look as follows in the Fourier basis:

$$F_z = \mathsf{P}_{\mathcal{D}_{\{z\}}} = \bigotimes_{i=1}^k \left(\frac{1}{M} \sum_{v,w \in Y} e^{\frac{2\pi\iota}{M}(w-v)\cdot y_i} |w\rangle\langle v|_{\mathcal{I}_{x_i}} \right), \tag{7.32}$$

and hence

$$\begin{aligned}
&F_z \mathsf{Comp}^\dagger \mathsf{P}_{\mathcal{D}_{\{z\}}} \mathsf{Comp} \\
&= \bigotimes_{i=1}^k \left(\frac{1}{M} \sum_{v,w \in Y} e^{\frac{2\pi\iota}{M}(w-v)\cdot y_i} |w\rangle\langle v|_{\mathcal{I}_{x_i}} \right) \bigotimes_{i=1}^k \left(\frac{1}{M} \sum_{v,w \in (Y \setminus \{0\})} e^{\frac{2\pi\iota}{M}(w-v)\cdot y_i} |w\rangle\langle v|_{\mathcal{I}_{x_i}} \right) \\
&= \bigotimes_{i=1}^k \left(\frac{M-1}{M^2} \sum_{\substack{v,w \in Y: \\ v \neq 0}} e^{\frac{2\pi\iota}{M}(w-v)\cdot y_i} |w\rangle\langle v|_{\mathcal{I}_{x_i}} \right).
\end{aligned} \tag{7.33}$$

We abbreviate $\boldsymbol{v} := (v_1, \ldots, v_k)$ and similarly introduce $|\boldsymbol{v}\rangle_{\mathcal{I}_{\boldsymbol{x}}} := \bigotimes_{i=1}^k |v_i\rangle_{\mathcal{I}_{x_i}}$. We also abbreviate

$$e^{\frac{2\pi\iota}{M}(\boldsymbol{w}-\boldsymbol{v})\cdot\boldsymbol{y}} := \prod_{i=1}^k e^{\frac{2\pi\iota}{M}(w_i - v_i)\cdot y_i}.$$

Using this new notation, we can apply both (7.32) and (7.33) to expand the expression

$F_z - F_z\mathsf{Comp}^\dagger\mathsf{P}_{\mathcal{D}_{\{z\}}}\mathsf{Comp}$ from (7.31) as

$$\bigotimes_{i=1}^{k}\left(\frac{1}{M}\sum_{v,w\in Y} e^{\frac{2\pi\iota}{M}(w-v)\cdot y_i}|w\rangle\langle v|_{\mathcal{I}_{x_i}}\right) - \bigotimes_{i=1}^{k}\left(\frac{M-1}{M^2}\sum_{\substack{v,w\in Y:\\ v\neq 0}} e^{\frac{2\pi\iota}{M}(w-v)\cdot y_i}|w\rangle\langle v|_{\mathcal{I}_{x_i}}\right)$$
$$= \frac{1}{M^k}\sum_{\boldsymbol{v},\boldsymbol{w}\in Y^k} e^{\frac{2\pi\iota}{M}(\boldsymbol{w}-\boldsymbol{v})\cdot\boldsymbol{y}}|\boldsymbol{w}\rangle\langle\boldsymbol{v}|_{\mathcal{I}_{\boldsymbol{x}}} - \left(\frac{M-1}{M^2}\right)^k \sum_{\substack{\boldsymbol{v},\boldsymbol{w}\in Y^k:\\ \nexists i: v_i=0}} e^{\frac{2\pi\iota}{M}(\boldsymbol{w}-\boldsymbol{v})\cdot\boldsymbol{y}}|\boldsymbol{w}\rangle\langle\boldsymbol{v}|_{\mathcal{I}_{\boldsymbol{x}}}$$
$$= \frac{1}{M^k}\sum_{\substack{\boldsymbol{v},\boldsymbol{w}\in Y^k\\ \exists i: v_i=0}} e^{\frac{2\pi\iota}{M}(\boldsymbol{w}-\boldsymbol{v})\cdot\boldsymbol{y}}|\boldsymbol{w}\rangle\langle\boldsymbol{v}|_{\mathcal{I}_{\boldsymbol{x}}} + \left(\frac{1}{M^k} - \left(\frac{M-1}{M^2}\right)^k\right)\sum_{\substack{\boldsymbol{v},\boldsymbol{w}\in Y^k:\\ \nexists i: v_i=0}} e^{\frac{2\pi\iota}{M}(\boldsymbol{w}-\boldsymbol{v})\cdot\boldsymbol{y}}|\boldsymbol{w}\rangle\langle\boldsymbol{v}|_{\mathcal{I}_{\boldsymbol{x}}}.$$

We now bound its norm by applying a counting argument on the number of $\boldsymbol{v},\boldsymbol{w}\in Y^k$ where either one or none of the v_i is equal to 0:

$$\begin{aligned}
&\left\|F_z - F_z\mathsf{Comp}^\dagger\mathsf{P}_{\mathcal{D}_{\{z\}}}\mathsf{Comp}\right\|^2\\
&\leq \frac{1}{M^{2k}}\left(M^{2k} - M^k(M-1)^k\right) + \left(\frac{1}{M^k} - \left(\frac{M-1}{M^2}\right)^k\right)^2 M^k(M-1)^k\\
&= 1 - \left(\frac{M-1}{M}\right)^k + \left(1 - \left(\frac{M-1}{M}\right)^k\right)^2\left(\frac{M-1}{M}\right)^k\\
&= 1 - 2\left(1-\frac{1}{M}\right)^{2k} + \left(1-\frac{1}{M}\right)^{3k} \leq 1 - \left(1-\frac{1}{M}\right)^{2k} \leq \frac{2k}{M}.
\end{aligned}$$

In the final inequality we have made use of the fact that $M > k \geq 1$, allowing us to apply Bernoulli's inequality:

$$\left(1-\frac{1}{M}\right)^{2k} \geq 1 - \frac{2k}{M}. \qquad \square$$

Knowing that $\frac{2k}{M}$ is a valid value for η, suppose that $\epsilon \leq 1 - (9-4\sqrt{2})\frac{k}{M}$. Then

$$\sqrt{1-\epsilon} - \left(2\sqrt{2}-1\right)\sqrt{\frac{k}{M}} \geq 0.$$

Together with Claim 7.4.2, (7.30) and Lemma 7.4.4, this means that our MLA matrix Γ satisfies (7.28), where in the penultimate step we use that $\|\Lambda_1\Pi_{\leq t}\mathcal{O}_{x,y}\Pi_{\leq t-1}\Lambda_0\|$ is monotonically non-decreasing in t (see Fact 7.3.6):

$$\begin{aligned}
&\sqrt{1-\epsilon} - \sqrt{\frac{k}{M}} \leq \sqrt{1-\epsilon} - \sqrt{\frac{k}{M}} + \sqrt{1-\epsilon} - (2\sqrt{2}-1)\sqrt{\frac{k}{M}} = 2\left(\sqrt{1-\epsilon} - \sqrt{\eta}\right)\\
&\leq \sum_{t=1}^{6T}\max_{x\in X, y\in Y}\|\Lambda_1\Pi_{\leq t}\mathcal{O}_{x,y}\Pi_{\leq t-1}\Lambda_0\| \leq \sum_{t=1}^{6T}\max_{x\in X, y\in Y}\|\Pi_{1,t}\mathcal{O}_{x,y}\Pi_{0,t-1}\|.
\end{aligned}$$

7.5 A Strong Direct Product Theorem

The machinery of MLA matrices seems a bit overcomplicated compared to what we actually needed in the reduction in Section 7.4. Since we only considered multiplicative matrices where $\ell = 1$, we obtain the "ladder" prefix automatically. We will need general ℓ however if we want to compute a function $\mathcal{F}$ on ℓ independent instances simultaneously.

Although it does not seem to fit in the framework of [CFHL21] directly, the compressed oracle framework also has the powerful property of being able to exhibit *strong direct product theorems* (SDPT), as shown in [LZ19a, HM23]. Such a theorem states that if we try to compute $\mathcal{F}$ on k independent inputs in fewer queries than k times the queries needed for a single instance of $\mathcal{F}$, then our success probability will decrease exponentially in k.

It was already shown by [Špa08] that the multiplicative adversary method directly satisfies a SDPT. Here we show that a similar proof as in [AMRR11], which is based on the proof in [Špa08], also holds for the MLA method due to the fact (which we will prove) that the set of MLA matrices is closed under tensor powers. This motivates the study of the MLA method as a simplification of the multiplicative adversary method, since it maintains the property of satisfying a SDPT.

We introduce the following notation for this section: for any problem $\mathcal{F} : \mathsf{Func} \to 2^{\Sigma}$ and integer $k \geq 1$ let $\mathcal{F}^{(k)} : \mathsf{Func}^k \to (2^{\Sigma})^k$ be defined as

$$\mathcal{F}^{(k)}(h_1, \ldots, h_k) = (\mathcal{F}(h_1), \ldots, \mathcal{F}(h_k)).$$

Theorem 7.5.1. *For any problem $\mathcal{F} : \mathsf{Func} \to 2^{\Sigma}$, input distribution σ on Func, and fixed $\eta \leq \frac{1}{2}$, let $\mathsf{MLADV}^{\sigma}_{1-\epsilon,\eta}(\mathcal{F})$ be the lower bound on $Q^{\sigma}_{\epsilon}(\mathcal{F})$ obtained by Corollary 7.3.5. Then there exists a constant $c \in (0,1)$ and integer $k > 361$ we have*

$$\mathsf{MLADV}^{\sigma^k}_{c^k,\eta^{\frac{2k}{5}}}(\mathcal{F}^{(k)}) \geq \frac{k}{10}\mathsf{MLADV}^{\sigma}_{1-\epsilon,\eta}(\mathcal{F}).$$

Proof. Let Γ, λ denote the optimal values in Corollary 7.3.5 for a fixed $\eta \leq \frac{1}{2}$. We use these to construct Γ' (with eigenspaces denoted by Λ'_j), λ' and η' for $\mathcal{F}^{(k)}$ as follows:

$$\Gamma' := \Gamma^{\otimes k}, \lambda' := \lambda^{\frac{k}{10}}, \eta' := \eta^{\frac{2k}{5}}.$$

This construction yields a positive definite matrix $\Gamma' \in \mathbb{C}^{\mathsf{Func}^k \times \mathsf{Func}^k}$ with smallest eigenvalue 1 of the form

$$\Gamma' = \Gamma^{\otimes k} = \sum_{j=0}^{k\cdot\ell} \kappa^j \Lambda'_j, \tag{7.34}$$

where

$$\Lambda'_j = \sum_{\substack{i_1,\ldots,i_k \in [j]_0: \\ i_1+\cdots+i_k=j}} \Lambda_{i_1} \otimes \cdots \otimes \Lambda_{i_k}.$$

We similarly define

$$\Pi'_{\leq t} = \sum_{\substack{t_1,\ldots,t_k \in [t]_0: \\ t_1+\cdots+t_k=t}} \Pi_{\leq t_1} \otimes \cdots \otimes \Pi_{\leq t_k},$$

where each $\Pi_t := \Pi_{\leq t} - \Pi_{\leq t-1}$ (as in the proof of Fact 7.3.6). We now set out to show that Γ' is of the correct form to use it as an upper bound for $\mathsf{MLADV}^{\sigma^k}_{c^k,\eta^{\frac{2k}{5}}}(\mathcal{F}^{(k)})$:

Lemma 7.5.2. *Let Γ be an MLA matrix for $\mathcal{F}$ (see Definition 7.3.2) with $|\sigma\rangle$ as a 1-eigenvector. Then for any non-negative integer k, $\Gamma^{\otimes k}$ is an MLA matrix for $\mathcal{F}^{(k)}$ with $|\sigma\rangle^{\otimes k}$ is a 1-eigenvector .*

Proof. By construction, $\Gamma' = \Gamma^{\otimes k}$ is already of the desired form (see (7.34) and Definition 7.3.2), and since $|\sigma\rangle$ is a 1-eigenvector of Γ, $|\sigma\rangle^{\otimes k}$ is a 1-eigenvector of Γ'. Additionally, since Γ commutes with each $\Pi_{\leq t}$, it follows that Γ' commutes with $\Pi'_{\leq t}$. What remains to verify is that Γ' satisfies (7.18).

Since $\mathsf{Func}^k = (Y^X)^k = (Y^{X'})$, where X' is a set of size kN, there exist unique x and k' for every $x' \in X'$ and $y \in Y$ such that

$$\mathcal{O}_{x',y} = I^{\otimes k'-1} \otimes \mathcal{O}_{x,y} \otimes I^{\otimes k-k'},$$

where I is the identity on $\mathbb{C}^{\mathsf{Func}\times\mathsf{Func}}$. Then, since Γ commutes with each Π_t, we can decompose $\left\|\Lambda'_{j'}\Pi'_{\leq t}\mathcal{O}_{x',y}\Pi'_{\leq t-1}\Lambda'_j\right\|$ as

$$\left\| \sum_{\substack{i_1,\ldots,i_k\in[j]_0,\\ i'_1,\ldots,i'_k\in[j']_0:\\ i_1+\cdots+i_k=j,\\ i'_1+\cdots+i'_k=j'}} \sum_{\substack{t_1,\ldots,t_k\in[t]_0,\\ t'_1,\ldots,t'_k\in[t]_0:\\ t_1+\cdots+t_k=t-1,\\ t'_1+\cdots+t'_k=t}} \underbrace{\Lambda_{i'_1}\Pi_{t'_1}\Pi_{t_1}\Lambda_{i_1}}_{=\delta_{i_1,i'_1}\delta_{t_1,t'_1}\Pi_{t_1}} \otimes\cdots\otimes \Lambda_{i'_{k'}}\Pi_{t'_{k'}}\mathcal{O}_{x,y}\Pi_{t_k}\Lambda_{i_{k'}} \otimes\cdots\otimes \underbrace{\Lambda_{i'_k}\Pi_{t'_k}\Pi_{t_k}\Lambda_{i_k}}_{=\delta_{i_k,i'_k}\delta_{t_k,t'_k}\Pi_{t_k}} \right\|$$

$$\leq \left\| \sum_{\substack{i\in[j]_0,i'\in[j']_0:\\ i'-i=j'-j}} \Lambda_{i'}\Pi_{\leq t}\mathcal{O}_{x,y}\Pi_{\leq t-1}\Lambda_i \right\| \leq \max_{\substack{i\in[j]_0,i'\in[j']_0:\\ i'-i=j'-j}} \left\|\Lambda_{i'}\Pi_{\leq t}\mathcal{O}_{x,y}\Pi_{\leq t-1}\Lambda_i\right\|. \tag{7.35}$$

In the last inequality, we have used the fact that each term $\Lambda_{i'}\Pi_{\leq t}\mathcal{O}_{x,y}\Pi_{\leq t-1}\Lambda_i$ in the sum has orthogonal images and coimages. It now follows from (7.35) that for every $x' \in X'$, $y \in Y$, $t \leq T$, and $j, j' \in [k\cdot\ell]_0$ with $|j-j'| > 1$, we have

$$\left\|\Lambda'_{j'}\Pi'_{\leq t}\mathcal{O}_{x',y}\Pi'_{\leq t-1}\Lambda'_j\right\| \leq \max_{\substack{x\in X,i\in[j]_0,i'\in[j']_0:\\ i'-i=j'-j}} \left\|\Lambda_{i'}\mathcal{O}_{x,y}\Lambda_i\right\| = 0, \tag{7.36}$$

since Γ is an MLA matrix and thus satisfies (7.18) itself and the fact that Γ commutes with both $\Pi_{\leq t-1}$ and $\Pi_{\leq t}$. □

To prove Theorem 7.5.1, we show that any T satisfying

$$1 + \left(\lambda' - 1\right)\left(\sqrt{c^k} - \sqrt{\eta'}\right)^2 \leq \prod_{t=1}^{T}\left(1 + \max_{\substack{j\in[k\cdot\ell-1]_0,\\ x'\in X^k, y\in Y}} \frac{\kappa-1}{\sqrt{\kappa}}\left\|\Lambda'_{j+1}\Pi'_{\leq t}\mathcal{O}'_{x,y}\Pi'_{\leq t-1}\Lambda'_j\right\|\right)^2, \tag{7.37}$$

also satisfies

$$1 + (\lambda - 1)\left(\sqrt{1-\epsilon} - \sqrt{\eta}\right)^2 \leq \prod_{t=1}^{(10/k)T}\left(1 + \max_{\substack{i\in[\ell-1]_0,\\ x\in X, y\in Y}} \frac{\kappa-1}{\sqrt{\kappa}}\left\|\Lambda_{i+1}\Pi_{\leq t}\mathcal{O}_{x,y}\Pi_{\leq t-1}\Lambda_i\right\|\right)^2. \tag{7.38}$$

The theorem then follows from Corollary 7.3.5. For the choices of λ' and η', we know by the assumptions in Theorem 7.3.4 that $\|F_z\Lambda_{\mathsf{bad}}\|^2 \leq \eta$ for every $z \in \Sigma$, where Λ_{bad} is the projector onto the eigenspaces of Γ corresponding to eigenvalues smaller than λ and $F_z = \sum_{f\in\mathsf{Func}:\mathcal{F}(f)\ni z} |f\rangle\langle f|$. Now let Λ'_{bad} be the projector onto the eigenspaces of Γ' corresponding to eigenvalues smaller than λ'. Abbreviate $\boldsymbol{z} = (z_1,\ldots,z_k) \in \Sigma^k$ and define $F_{\boldsymbol{z}} = \bigotimes_{j=1}^k F_{z_j}$. Let V_{bad} denote the space that Λ_{bad} projects onto, let V_{good} be its orthogonal complement and analogously define V'_{bad}. By construction of $\Gamma' = \Gamma^{\otimes k}$, we know that Λ'_{bad} is a subspace of the direct sum of spaces $\bigotimes_{j=1}^k V_{v_j}$ where $\boldsymbol{v} = (v_1,\ldots,v_k) \in \{\mathsf{good},\mathsf{bad}\}^k$. Since all the eigenvalues of Γ are bounded below by 1 and V'_{bad} is the direct sum of all eigenspaces of Γ' with eigenvalue smaller than $\lambda' = \lambda^{k/10}$, it must be that

the number of good subspaces, denoted by $|\boldsymbol{v}|$, is at most $k/10$. This means that we can decompose any state $|\phi\rangle \in V'_{\mathsf{bad}}$ as a product state

$$|\phi\rangle = \sum_{\substack{\boldsymbol{v}\in\{\text{good, bad}\}^k:\\ |\boldsymbol{v}|\leq\frac{k}{10}}} \alpha_{\boldsymbol{v}}|\phi_{\boldsymbol{v}}\rangle = \sum_{\substack{\boldsymbol{v}\in\{\text{good, bad}\}^k:\\ |\boldsymbol{v}|\leq\frac{k}{10}}} \alpha_{\boldsymbol{v}} \bigotimes_{j=1}^{k} |\phi_{v_j}\rangle,$$

where $|\phi_{v_j}\rangle \in V_{v_j}$. It now follows, as in [Špa08], whenever $\eta \leq 1/2$ and $k \geq 361$ for every $\boldsymbol{z} \in \Sigma^k$ that there exists $|\phi\rangle \in V'_{\mathsf{bad}}$ such that for every $\boldsymbol{z} \in \Sigma^k$

$$\begin{aligned}\left\|F_{\boldsymbol{z}}\Lambda'_{\mathsf{bad}}\right\|^2 &= \Big\| \sum_{\substack{\boldsymbol{v}\in\{\text{good, bad}\}^k:\\ |\boldsymbol{v}|\leq\frac{k}{10}}} \alpha_{\boldsymbol{v}} F_{\boldsymbol{z}}|\phi_{\boldsymbol{v}}\rangle\Big\| \leq \Big\| \sum_{\substack{\boldsymbol{v}\in\{\text{good, bad}\}^k:\\ |\boldsymbol{v}|\leq\frac{k}{10}}} \bigotimes_{j=1}^{k} F_{z_j}|\phi_{v_j}\rangle\Big\| \\ &\leq \eta^{\frac{9k}{10}} \sum_{\substack{\boldsymbol{v}\in\{\text{good, bad}\}^k:\\ |\boldsymbol{v}|\leq\frac{k}{10}}} \leq \eta^{\frac{9k}{10}} \sum_{i=0}^{k/10} \binom{k}{i} \leq k\binom{k}{k/10}\eta^{\frac{9k}{10}} \leq k(10e)^{k/10}\eta^{\frac{9k}{10}}.\end{aligned}$$

Under the assumptions of the lemma, we know that $\eta \leq \frac{1}{2}$ and $k \geq 361$, meaning

$$k(10e)^{k/10}\eta^{\frac{9k}{10}} \leq 2^{k/2}\eta k/2\eta^{2k/5} \leq \eta^{2k/5} = \eta'.$$

To finalise the proof, note that for any fixed $\eta < 1/2, \epsilon \in (0, 1-\eta)$, and $\lambda > 1$ we have

$$\frac{1+(\lambda-1)\left(\sqrt{1-\epsilon}-\sqrt{\eta}\right)^2}{\lambda} < \frac{1+(\lambda-1)}{\lambda} = 1.$$

Hence, there exists a constant $c \in (0,1)$ such that for every $k \geq 361$ we have

$$\left(\frac{1+(\lambda-1)\left(\sqrt{1-\epsilon}-\sqrt{\eta}\right)^2}{\lambda}\right)^{\frac{k}{10}} + \eta^{\frac{k}{5}} \leq c^{\frac{k}{2}}. \tag{7.39}$$

Therefore, given our choices of λ' and η', we find

$$\begin{aligned}1+\left(\lambda'-1\right)\left(\sqrt{c^k}-\sqrt{\eta'}\right)^2 &\geq 1+\left(\lambda'-1\right)\left(\frac{1+(\lambda-1)\left(\sqrt{1-\epsilon}-\sqrt{\eta}\right)^2}{\lambda}\right)^{\frac{k}{10}} \\ &= 1+\left(1-\lambda^{-\frac{k}{10}}\right)\left(1+(\lambda-1)\left(\sqrt{1-\epsilon}-\sqrt{\eta}\right)^2\right)^{\frac{k}{10}} \\ &\geq \left(1+(\lambda-1)\left(\sqrt{1-\epsilon}-\sqrt{\eta}\right)^2\right)^{\frac{k}{10}},\end{aligned} \tag{7.40}$$

where the final inequality is due to the fact that $1+(\lambda-1)\left(\sqrt{1-\epsilon}-\sqrt{\eta}\right)^2 \leq \lambda$ by (7.39).

We can conclude the theorem by showing that if (7.37) holds, then so must (7.38), by combining (7.35) with (7.36) and (7.40) and the fact that $\|\Lambda_{i+1}\Pi_{\leq t}\mathcal{O}_{x,y}\Pi_{\leq t-1}\Lambda_i\|$ is

monotonically non-decreasing in t (see Fact 7.3.6):

$$1+(\lambda-1)\left(\sqrt{1-\epsilon}-\sqrt{\eta}\right)^2 \leq \left(1+\left(\lambda'-1\right)\left(\sqrt{c^k}-\sqrt{\eta'}\right)^2\right)^{\frac{10}{k}}$$
$$\leq \left(\prod_{t=1}^{T}\left(1+\max_{\substack{j\in[k\cdot\ell-1]_0,\\ x'\in X^k,y\in Y}}\frac{\kappa-1}{\sqrt{\kappa}}\left\|\Lambda'_{j+1}\Pi'_{\leq t}\mathcal{O}_{x',y}\Pi'_{\leq t-1}\Lambda'_j\right\|\right)^2\right)^{\frac{10}{k}}$$
$$\leq \left(\prod_{t=1}^{T}\left(1+\max_{\substack{i\in[\ell-1]_0,\\ x\in X,y\in Y}}\frac{\kappa-1}{\sqrt{\kappa}}\left\|\Lambda_{i+1}\Pi_{\leq t}\mathcal{O}_{x,y}\Pi_{\leq t-1}\Lambda_i\right\|\right)^2\right)^{\frac{10}{k}}$$
$$\leq \prod_{t=1}^{(10/k)T}\left(1+\max_{\substack{i\in[\ell-1]_0,\\ x\in X,y\in Y}}\frac{\kappa-1}{\sqrt{\kappa}}\left\|\Lambda_{i+1}\Pi_{\leq t}\mathcal{O}_{x,y}\Pi_{\leq t-1}\Lambda_i\right\|\right)^2. \qquad \square$$

7.6 Inverting permutations

In this section, we show that the approach in [Ros21] to generalise the compressed oracle framework to permutations is also captured by the multiplicative ladder adversary (MLA) method. Since in the setting of [Ros21] we are working with random permutations, rather than random functions, we consider Perm: the set of all permutations from X to X, where $X=[N-1]_0$. Our objective is to find the unique preimage of 0 under a permutation f, meaning $\mathcal{F}:\mathsf{Perm}\to X$, where $\mathcal{F}(f)=x$ if and only if $f(x)=0$. Note that here, we can revert to the original notion of "success" (see Definition 7.1.4 and Definition 7.2.9). We aim to apply Corollary 7.3.5 to recover the following result:

Theorem 7.6.1 (Corollary 5 in [Ros21]). *Let* Perm *be the set of all permutations from X to X, where $X=[N-1]_0$ and let $\mathcal{F}:\mathsf{Perm}\mapsto X$, where $\mathcal{F}(f)=x$ if and only if $f(x)=0$. Any T-query quantum algorithm $\mathcal{A}$ successfully outputs $\mathcal{F}(f)$ when f is drawn uniformly from* Perm *with success probability at most $\left(1+2\sqrt{2}T\right)^2/(N-4T)$.*

We apply Corollary 7.3.5 by constructing an MLA matrix Γ from the constructions in [Ros21]. In the permutation case, the states that make up $\mathsf{Space}_t(|\sigma\rangle)$ (see (7.10)) are (for any $t\in[N]_0$):

$$|v^{y_1,\ldots,y_t}_{x_1,\ldots,x_t}\rangle := \frac{1}{\sqrt{(N-t)!}}\sum_{\substack{f\in\mathsf{Perm}\\ \forall i\in[t]:f(x_i)=y_i}}|f\rangle.$$

Each such state can be interpreted as the database $|D\rangle$ from Section 7.2.3, where D contains the input-output pairs $(x_1,y_1),\ldots,(x_t,y_t)$. In [Ros21], the span of these states is denoted by A_t:

$$A_t := \mathsf{Space}_t(|\sigma\rangle)=\mathsf{span}\left\{|v^{y_1,\ldots,y_t}_{x_1,\ldots,x_t}\rangle:((x_1,y_1),\ldots,(x_t,y_t))\in(X\times X)^t\right\}.$$

The second space introduced in [Ros21], where $t\in[N]$, is

$$B_t := \mathsf{span}\left\{|v^{0,\ldots,y_t}_{x_1,\ldots,x_t}\rangle:((x_1,0),(x_2,y_2),\ldots,(x_t,y_t))\in(X\times X)^t\right\}\subseteq A_t, \tag{7.41}$$

where a preimage of zero is captured in the database. We have already seen in (7.15) that $A_t\subseteq A_{t+1}$. Instead of summing over the different possible y values of the new input-output pair, we can also sum over the possible x values:

$$|v^{y_1,\ldots,y_t}_{x_1,\ldots,x_t}\rangle := \sqrt{N-(k+1)}\sum_{x\in X\setminus\{x_1,\ldots,x_t\}}|v^{y_1,\ldots,y_t,y}_{x_1,\ldots,x_t,x}\rangle,$$

where y is any fixed element in $Y \setminus \{y_1, \dots, y_t\}$. By choosing $y = 0$, we actually obtain for every $t \in [N]$ that

$$A_{t-1} \subseteq B_t \subseteq A_t. \tag{7.42}$$

With these spaces, we can construct our MLA matrix Γ. Although it seems reasonable to let the eigenspaces of Γ correspond to the spaces A_t and B_t, (7.42) shows that these spaces are not orthogonal. We address this by introducing the projectors $\widehat{\Pi}_{1,t}$ and $\widehat{\Pi}_{0,t}$, which project onto $\bigoplus_{i=1}^{t} \left(B_i \cap (A_{i-1})^\perp\right)$ and $\bigoplus_{i=1}^{t} \left(A_i \cap (B_i)^\perp\right)$, respectively. In [Ros21], these projectors are called $\widehat{\Pi}_t^{\text{high}}$ and $\widehat{\Pi}_t^{\text{low}}$. To understand the intuition as to why these spaces are considered, we refer the reader to [Ros21]. For now, we can think of the projectors $\widehat{\Pi}_{1,t}$ and $\widehat{\Pi}_{0,t}$ as the permutation counterparts of $\Pi_{1,t}$ and $\Pi_{0,t}$ that we defined in (7.26). Once again, our MLA matrix will be of the form $\Gamma = \Lambda_0 + \kappa\Lambda_1$, where we set $\Lambda_1 = \widehat{\Pi}_{1,N}$, and accordingly set $\Lambda_0 = I - \Lambda_1$:

Claim 7.6.2. *Let $\Lambda_1 \coloneqq \widehat{\Pi}_{1,N}$, which projects onto $\bigoplus_{i=1}^{N} \left(B_i \cap (A_{i-1})^\perp\right)$, let $\Lambda_0 = I - \Lambda_1$ and $\Gamma = \Lambda_0 + \kappa\Lambda_1$ for some constant $\kappa > 1$. Then Γ is an MLA matrix as defined in Definition 7.3.2 with $|\sigma\rangle$ as a 1-eigenvector.*

Proof. It is clear that this construction makes Γ positive definite with smallest eigenvalue 1 and largest eigenvalue κ. Additionally, by (7.42) each component of the direct sum $\bigoplus_{i=1}^{N} \left(B_i \cap (A_{i-1})^\perp\right)$ is a subspace of A_N, meaning Λ_1 (and hence also Γ by construction) commutes with $\Pi_{\leq N}$ (which projects onto A_N) and hence also with every $\Pi_{\leq t} \preceq \Pi_{\leq N}$. Lastly, since $\ell = 1$, we automatically satisfy (7.18) from Definition 7.3.2, meaning we only need to verify that $|\sigma\rangle = \frac{1}{\sqrt{N!}} \sum_{f \in \mathsf{Perm}} |f\rangle$ is indeed an eigenvector of Γ with eigenvalue 1. Recall from (7.42) that $A_{t-1} \subseteq A_t$ for $t \in [N]$. In particular, this means that $|\sigma\rangle \in A_0$ is orthogonal to each $(A_t)^\perp$ for $t \in [N]_0$ and therefore in particular also to the direct sum $\bigoplus_{t=1}^{N} B_t \cap (A_{t-1})^\perp$, meaning

$$\Gamma|\sigma\rangle = |\sigma\rangle - \widehat{\Pi}_{1,N}|\sigma\rangle = |\sigma\rangle. \qquad \square$$

By choosing $\lambda = \kappa = 1 + (e-1)/\left(\sqrt{1-\epsilon} - \sqrt{\eta}\right)^2$ (as seen in (7.30)), Corollary 7.3.5 now tells us that $Q_{1-\epsilon}(\mathcal{F})$ is lower bounded by the smallest T satisfying

$$\sqrt{1-\epsilon} - \sqrt{\eta} \leq 2\sqrt{2} \sum_{t=1}^{T} \max_{x \in X, y \in Y} \left\| \Lambda_1 \Pi_{\leq t} \mathcal{O}_{x,y} \Pi_{\leq t-1} \Lambda_0 \right\|. \tag{7.43}$$

To be able to continue, we first need to show that Claim 7.4.2 also holds in the case of permutations:

Claim 7.6.3. *For each $t \in [N]_0$, we have*

$$\Pi_{\leq t} \Lambda_0 = \widehat{\Pi}_{0,t}, \qquad \Lambda_1 \Pi_{\leq t} = \widehat{\Pi}_{1,t}. \tag{7.44}$$

Proof. Both parts of the claim follow from the fact that $A_{t-1} \subseteq B_t \subseteq A_t$ (see (7.42)): Starting with Λ_1, we know that $\Lambda_1 \Pi_{\leq t}$ projects onto

$$A_t \cap \bigoplus_{i=1}^{N} \left(B_i \cap (A_{i-1})^\perp\right) = A_t \cap \bigoplus_{i=1}^{t} \left(B_i \cap (A_{i-1})^\perp\right) = \bigoplus_{i=1}^{t} \left(B_i \cap (A_{i-1})^\perp\right),$$

which is the space that $\widehat{\Pi}_{1,t}$ projects onto. Similarly, we obtain that $\Pi_{\leq t}\Lambda_0$ projects onto

$$A_t \cap \left(\bigoplus_{i=1}^{N} \left(B_i \cap (A_{i-1})^\perp\right)\right)^\perp = A_t \cap \bigcap_{i=1}^{N} \left(A_{i-1} \cup (B_i)^\perp\right) = \bigoplus_{i=1}^{t} \left(A_i \cap (B_{i-1})^\perp\right),$$

which is the space that $\widehat{\Pi}_{0,t}$ projects onto. $\square$

By Claim 7.6.3, we can relate the right-hand side of (7.43) to the projectors $\widehat{\Pi}_{0,t}, \widehat{\Pi}_{1,t}$ via the inequality

$$\|\Lambda_1 \Pi_{\leq t} \mathcal{O}_{x,y} \Pi_{\leq t-1} \Lambda_0\| = \left\| \widehat{\Pi}_{1,t} \mathcal{O}_{x,y} \widehat{\Pi}_{0,t-1} \right\|. \tag{7.45}$$

It is shown in Claim 11 and Claim 12 in [Ros21] that

$$\left\| \widehat{\Pi}_{1,t} \mathcal{O}_{x,y} \widehat{\Pi}_{0,t-1} \right\| \leq \frac{2\sqrt{2}}{\sqrt{N-4t}}.$$

By plugging this into (7.43), together with (7.45), we obtain

$$\sqrt{1-\epsilon} - \sqrt{\eta} \leq 2\sqrt{2} \sum_{t=1}^{T} \max_{x \in X, y \in Y} \frac{2\sqrt{2}}{\sqrt{N-4t}} \leq \frac{8T}{\sqrt{N-4T}}. \tag{7.46}$$

The last step in proving Theorem 7.6.1 consists of finding a valid value for η, meaning we have to bound $\|F_z \Lambda_{\mathsf{bad}}\|$. By construction of Γ and our choice of λ, the projection Λ_{bad} is equal to Λ_0. The final piece of the puzzle can again be found in [Ros21], this time in Claim 10, where it is shown that

$$\|F_z \Lambda_0\| \leq \frac{1}{\sqrt{N-2T}}.$$

Combining this with (7.46), results in an upper bound on our success probability of

$$\left(\frac{8T}{\sqrt{N-4T}} + \frac{1}{\sqrt{N-2T}} \right)^2 \leq \frac{(1+8T)^2}{N-4T},$$

which recovers Theorem 7.6.1 up to a constant factors.

Bibliography

[AC21] Yosi Atia and Shantanav Chakraborty. Improved upper bounds for the hitting times of quantum walks. *Physical Review A*, 104:032215, 2021. arXiv: 2005.04062

[AdW01] Andris Ambainis and Ronald de Wolf. Average-case quantum query complexity. *Journal of Physics A: Mathematical and General*, 34(35):6741, 2001. arXiv: quant-ph/9904079

[ADW17] Srinivasan Arunachalam and Ronald De Wolf. Optimizing the number of gates in quantum search. *Quantum Information & Computation*, 17(3-4):251–261, 2017. arXiv: 1512.07550

[AGJ20] Simon Apers, András Gilyén, and Stacey Jeffery. A unified framework of quantum walk search. In *Proceedings of the 38th Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 6:1–6:13, 2020. arXiv: 1912.04233

[AGJK20] Andris Ambainis, András Gilyén, Stacey Jeffery, and Martins Kokainis. Quadratic speedup for finding marked vertices by quantum walks. In *Proceedings of the 52nd ACM Symposium on the Theory of Computing (STOC)*, page 412–424, 2020. arXiv: 1903.07493

[Ajt05] Miklós Ajtai. A non-linear time lower bound for Boolean branching programs. *Theory of Computing*, 1:149–176, 2005.

[Amb00] Andris Ambainis. Quantum lower bounds by quantum arguments. In *Proceedings of the 32nd ACM Symposium on the Theory of Computing (STOC)*, pages 636–643, 2000. arXiv: quant-ph/0002066

[Amb04] Andris Ambainis. Quantum walk algorithm for element distinctness. In *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 22–31, 2004. arXiv: quant-ph/0311001

[Amb07] Andris Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007. Earlier version in FOCS'04. arXiv: quant-ph/0311001

[Amb10a] Andris Ambainis. A new quantum lower bound method, with an application to a strong direct product theorem for quantum search. *Theory of Computing*, 6(1):1–25, 2010. arXiv: quant-ph/0508200

[Amb10b] Andris Ambainis. Quantum search with variable times. *Theory of Computing Systems*, 47:786–807, 2010. arXiv: quant-ph/0609168

[AMRR11] Andris Ambainis, Loïck Magnin, Martin Roetteler, and Jérémie Roland. Symmetry-assisted adversaries for quantum state generation. In *2011 IEEE 26th Annual Conference on Computational Complexity*, pages 167–177. IEEE, 2011. arXiv: 1012.2112

[AP22] Simon Apers and Stephen Piddock. Elfs, trees and quantum walks. *arXiv preprint arXiv:2211.16379*, 2022. arXiv: 2211.16379

[AS04] Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51(4):595–605, 2004. arXiv: quant-ph/0112086

[AS19] Simon Apers and Alain Sarlette. Quantum fast-forwarding: Markov chains and graph property testing. *Quantum Information and Computation*, 19(3&4):181–213, 2019. arXiv: 1804.02321

[AŠdW06] Andris Ambainis, Robert Špalek, and Ronald de Wolf. A new quantum lower bound method, with applications to direct product theorems and time-space tradeoffs. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of Computing*, pages 618–633, 2006. arXiv: quant-ph/0511200

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997. arXiv: quant-ph/9701001

[BBC+01] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. Earlier version in FOCS'98. arXiv: quant-ph/9802049

[BCJ+13] Aleksandrs Belovs, Andrew M. Childs, Stacey Jeffery, Robin Kothari, and Frédéric Magniez. Time-efficient quantum walks for 3-distinctness. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 105–122, 2013. arXiv: 1302.7316

[BDH+05] Harry Buhrman, Christoph Dürr, Mark Heiligman, Peter Høyer, Frédéric Magniez, Miklos Santha, and Ronald de Wolf. Quantum algorithms for element distinctness. *SIAM Journal on Computing*, 34(6):1324–1330, 2005. Earlier version in CCC'01. arXiv: quant-ph/0007016

[BDPVA07] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge functions. In *ECRYPT hash workshop*, number 9, 2007.

[Bel12a] Aleksandrs Belovs. Learning-graph-based quantum algorithm for k-distinctness. In *Proceedings of the 53rd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 207–216, 2012. arXiv: 1205.1534

[Bel12b] Aleksandrs Belovs. Span programs for functions with constant-sized 1-certificates. In *Proceedings of the 44th ACM Symposium on the Theory of Computing (STOC)*, pages 77–84, 2012. arXiv: 1105.4024

[Bel13] Aleksandrs Belovs. Quantum walks and electric networks. arXiv: 1302.3143, 2013.

[BHMT02] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Quantum Information: A Millennium Volume*, volume 305 of *Contemporary Mathematics Series*, pages 53–74. AMS, 2002. arXiv: quant-ph/0005055

[BHT97] Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum algorithm for the collision problem. *ACM SIGACT News*, 28:14–19, 1997. arXiv: quant-ph/9705002

[BKT18] Mark Bun, Robin Kothari, and Justin Thaler. The polynomial method strikes back: Tight quantum query bounds via dual polynomials. In *Proceedings of the 50th ACM Symposium on the Theory of Computing (STOC)*, 2018. arXiv: 1710.09079

[BLH23] Shankar Balasubramanian, Tongyang Li, and Aram Harrow. Exponential speedups for quantum walks in random hierarchical graphs. *arXiv preprint arXiv:2307.15062*, 2023. arXiv: 2307.15062

[BLPS22] Harry Buhrman, Bruno Loff, Subhasree Patro, and Florian Speelman. Limits of quantum speed-ups for computational geometry and other problems: Fine-grained complexity via quantum walks. arXiv: 2106.02005, 2022.

[BR17] Aleksandrs Belovs and Ansis Rosmanis. Adversary lower bounds for the collision and the set equality problems. *Quantum Information and Computation*, 2017. arXiv: 1310.5185

[BS04] Howard Barnum and Michael Saks. A lower bound on the quantum query complexity of read-once functions. *Journal of Computer and System Sciences*, 69(2):244–258, 2004. arXiv: quant-ph/0201007

[BŠ06] Harry Buhrman and Robert Špalek. Quantum verification of matrix products. In *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 880–889, 2006. arXiv: quant-ph/0409035

[BV97] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26:1411–1473, 1997.

[CCD+03] Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the 35th ACM Symposium on the Theory of Computing (STOC)*, pages 59–68, 2003. arXiv: quant-ph/0209131

[CE05] Andrew M Childs and Jason M Eisenberg. Quantum algorithms for subset finding. 2005.

[CFHL21] Kai-Min Chung, Serge Fehr, Yu-Hsuan Huang, and Tai-Ning Liao. On the compressed-oracle technique, and post-quantum security of proofs of sequential work. In *Advances in Cryptology–EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part II*, pages 598–629. Springer, 2021. ePrint: 2020/1305

[CJKM13] Andrew M. Childs, Stacey Jeffery, Robin Kothari, and Frédéric Magniez. A time-efficient quantum walk for 3-distinctness using nested updates. arXiv: 1302.7316, 2013.

[CJOP20] Arjan Cornelissen, Stacey Jeffery, Maris Ozols, and Alvaro Piedrafita. Span programs and quantum time complexity. In *Proceedings of the 45th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 21:1–26:14, 2020. arXiv: 2005.01323

[CLN16] Craig Costello, Patrick Longa, and Michael Naehrig. Efficient algorithms for supersingular isogeny diffie-hellman. In *Advances in Cryptology (CRYPTO 2016)*, pages 572–601, 2016. ePrint: 2016/413

[CMSZ19] Jan Czajkowski, Christian Majenz, Christian Schaffner, and Sebastian Zur. Quantum lazy sampling and game-playing proofs for quantum indifferentiability. *arXiv preprint arXiv:1904.11477*, 2019. arXiv: 1904.11477

[Cop02] Don Coppersmith. An approximate fourier transform useful in quantum factoring. *arXiv preprint quant-ph/0201067*, 2002. arXiv: quant-ph/0201067

[CRR+96] Ashok K. Chandra, Prabhakar Raghavan, Walter L. Ruzzo, Roman Smolensky, and Prasoon Tiwari. The electrical resistance of a graph captures its commute and cover times. *Computational Complexity*, 6(4):312–340, 1996.

[DFMS22] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Online-extractability in the quantum random-oracle model. In *Advances in Cryptology–EUROCRYPT 2022: 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30–June 3, 2022, Proceedings, Part III*, pages 677–706. Springer, 2022. ePrint: 2021/280

[DHHM06] Christoph Dürr, Mark Heiligman, Peter Høyer, and Mehdi Mhalla. Quantum query complexity of some graph problems. *SIAM Journal on Computing*, 35(6):1310–1328, 2006. Earlier version in ICALP'04. arXiv: quant-ph/0401091

[DT07] Sebastian Dörn and Thomas Thierauf. The quantum query complexity of algebraic properties. In *Fundamentals of Computation Theory: 16th International Symposium, FCT 2007, Budapest, Hungary, August 27-30, 2007. Proceedings 16*, pages 250–260. Springer, 2007. arXiv: 0705.1446

[dW19] Ronald de Wolf. Quantum computing: Lecture notes. *arXiv preprint arXiv:1907.09415*, 2019. arXiv: 1907.09415

[EPR35] Albert Einstein, Boris Podolsky, and Nathan Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical review*, 47(10):777, 1935.

[GHHM21] Alex B Grilo, Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. Tight adaptive reprogramming in the qrom. In *Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part I 27*, pages 637–667. Springer, 2021. arXiv: 2010.15103

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th ACM Symposium on the Theory of Computing (STOC)*, pages 212–219, 1996. arXiv: quant-ph/9605043

[Gro02] Lov K Grover. Trade-offs in the quantum search algorithm. *Physical Review A*, 66(5):052314, 2002.

[HHL09] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009. arXiv: 0811.3171

[HLŠ07] Peter Høyer, Troy Lee, and Robert Špalek. Negative weights make adversaries stronger. In *Proceedings of the 39th ACM Symposium on the Theory of Computing (STOC)*, pages 526–535, 2007. arXiv: quant-ph/0611054

[HM23] Yassine Hamoudi and Frédéric Magniez. Quantum time–space tradeoff for finding multiple collision pairs. *ACM Transactions on Computation Theory*, 15(1-2):1–22, 2023. arXiv: 2002.08944

[Jef14] Stacey Jeffery. *Frameworks for Quantum Algorithms*. PhD thesis, University of Waterloo, 2014. Available at http://uwspace.uwaterloo.ca/handle/10012/8710.

[JLR11] Svante. Janson, Tomasz. Luczak, and Andrzej Rucinski. *Random Graphs*. Wiley Series in Discrete Mathematics and Optimization. John Wiley & Sons, 2011.

[JZ] Stacey Jeffery and Sebastian Zur. The compressed oracle is a worthy adversary. *Under preparation*.

[JZ23] Stacey Jeffery and Sebastian Zur. Multidimensional quantum walks. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1125–1130, 2023. arXiv: 2208.13492

[Kit96] Alexei Y. Kitaev. Quantum measurements and the Abelian stabilizer problem. *ECCC*, TR96-003, 1996.

[Kit99] Alexei Kitaev. Quantum np. *Talk at AQIP*, 99, 1999.

[KNR09] Eyal Kaplan, Moni Naor, and Omer Reingold. Derandomized constructions of k-wise (almost) independent permutations. *Algorithmica*, 55(1):113–133, 2009.

[Li23] Jianqiang Li. Exponential speedup of quantum algorithms for the pathfinding problem. *arXiv preprint arXiv:2307.12492*, 2023. arXiv: 2307.12492

[LLL24] Guanzhong Li, Lvzhou Li, and Jingquan Luo. Recovering the original simplicity: succinct and deterministic quantum algorithm for the welded tree problem. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2454–2480. SIAM, 2024. arXiv: 2304.08395

[Llo96] Seth Lloyd. Universal quantum simulators. *Science*, 273(5278):1073–1078, 1996.

[LMR+11] Troy Lee, Rajat Mittal, Ben W. Reichardt, Robert Špalek, and Mário Szegedy. Quantum query complexity of state conversion. In *Proceedings of the 52nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 344–353, 2011. arXiv: 1011.3020

[LP16] Russell Lyons and Yuval Peres. *Probability on Trees and Networks*, volume 42 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, New York, 2016. Available at https://rdlyons.pages.iu.edu/.

[LR13] Troy Lee and Jérémie Roland. A strong direct product theorem for quantum query complexity. *computational complexity*, 22:429–462, 2013. arXiv: 1104.4468

[LZ19a] Qipeng Liu and Mark Zhandry. On finding quantum multi-collisions. In *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part III 38*, pages 189–218. Springer, 2019. ePrint: 2018/1096

[LZ19b] Qipeng Liu and Mark Zhandry. Revisiting post-quantum fiat-shamir. In *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39*, pages 326–355. Springer, 2019. ePrint: 2019/262

[LZ23] Jianqiang Li and Sebastian Zur. Multidimensional electrical networks and their application to exponential speedups for graph problems. *arXiv preprint arXiv:2311.07372*, 2023. arXiv: 2311.07372

[MMW24] Christian Majenz, Giulio Malavolta, and Michael Walter. Permutation superposition oracles for quantum query lower bounds. *arXiv preprint arXiv:2407.09655*, 2024. arXiv: 2407.09655

[MNRS11] Frédéric Magniez, Ashwin Nayak, Jérémie Roland, and Miklos Santha. Search via quantum walk. *SIAM Journal on Computing*, 40(1):142–164, 2011. Earlier version in STOC'07. arXiv: quant-ph/0608026

[MR15] Loïck Magnin and Jérémie Roland. Explicit relation between all lower bound techniques for quantum query complexity. *International Journal of Quantum Information*, 13(04):1350059, 2015. arXiv: 1209.2713

[MTZ20] Nikhil S. Mande, Justin Thaler, and Shuchen Zhu. Improved Approximate Degree Bounds for k-Distinctness. In *Proceedings of the 15th Conference on the Theory of Quantum Computation, Communication, and Cryptography (TQC)*, volume 158, pages 2:1–2:22, 2020. arXiv: 2002.08389

[NC00] Michael A. Nielsen and Isaac L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2000.

[Par70] James L Park. The concept of transition in quantum mechanics. *Foundations of physics*, 1(1):23–33, 1970.

[Pid19] Stephen Piddock. Quantum walk search algorithms and effective resistance. arXiv: 1912.04196, 2019.

[Rei09] Ben W Reichardt. Span programs and quantum query complexity: The general adversary bound is nearly tight for every boolean function. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 544–551. IEEE, 2009. arXiv: 0904.2759

[Ros21] Ansis Rosmanis. Tight bounds for inverting permutations via compressed oracle arguments. *arXiv preprint arXiv:2103.08975*, 2021. arXiv: 2103.08975

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. arXiv: quant-ph/9508027

[Sie86] William McC Siebert. *Circuits, signals, and systems*. MIT press, 1986.

[Špa08] Robert Špalek. The multiplicative quantum adversary. In *2008 23rd Annual IEEE Conference on Computational Complexity*, pages 237–248. IEEE, 2008. arXiv: quant-ph/0703237

[ST14] Daniel A Spielman and Shang-Hua Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM Journal on Matrix Analysis and Applications*, 35(3):835–885, 2014. arXiv: cs/0607105

[Sze04] Mario Szegedy. Quantum speed-up of Markov chain based algorithms. In *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 32–41, 2004. arXiv: quant-ph/0401053

[Tan09] Seiichiro Tani. Claw finding algorithms using quantum walk. *Theoretical Computer Science*, 410(50):5285–5297, 2009. arXiv: 0708.2584

[Vis13] Nisheeth K Vishnoi. Lx= b. *Foundations and Trends® in Theoretical Computer Science*, 8(1–2):1–141, 2013. https://www.cs.yale.edu/homes/vishnoi/Lxb-Web.pdf.

[Zha05] Shengyu Zhang. On the power of ambainis lower bounds. *Theoretical Computer Science*, 339(2-3):241–256, 2005. arXiv: quant-ph/0311060

[Zha19] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In *Annual International Cryptology Conference*, pages 239–268. Springer, 2019. ePrint: 2018/276

Abstract

Quantum computing holds the promise of solving certain problems far more efficiently than classical computing, but understanding its power and limitations requires grappling with concepts that are fundamentally different from classical algorithms. This dissertation aims to provide a structured approach to understanding quantum algorithms by focusing on frameworks that retain a connection to classical intuition and concepts, advancing our understanding of what quantum algorithms can—and cannot—achieve.

In Part I, we explore the capabilities of a specific class of quantum algorithms known as *quantum walks*. These quantum walks, the quantum analogs of classical random walks, serve as powerful yet accessible tools for solving a variety of computational problems. Designing a quantum walk algorithm typically begins with a classical random walk algorithm, which is then adapted into a quantum version with improved runtime performance. Our key contribution is the development of a novel way to adapt the underlying classical random walk, resulting in *multidimensional quantum walks*, a new class of quantum walks that significantly broadens the scope of existing techniques. By constructing a multidimensional quantum walk for the k-distinctness problem, we achieve a time-efficient solution that matches the best-known query upper bound up to polylogarithmic factors. Furthermore, by applying multidimensional quantum walks to the welded tree problem, we demonstrate that this new class of quantum walks is capable of achieving exponential speedups, making it the first instance of a (discrete) quantum walk to do so. Additionally, we extend the known connection between quantum walks and electrical networks to the multidimensional setting, introducing the concept of *multidimensional electrical networks*, which provide deeper insights into the underlying structure of quantum walk algorithms.

In Part II, we turn to the limitations of quantum algorithms by examining techniques for establishing quantum query lower bounds. These lower bounds represent the minimum computational cost required for any quantum algorithm to solve a given computational problem, regardless of the quantum algorithm used. We investigate *the compressed oracle technique*, a method renowned for deriving strong, yet intuitive quantum query lower bounds. However, this technique has historically been restricted to a limited range of problems. To overcome this limitation, we introduce the *multiplicative ladder adversary method*, a simplified and more intuitive adaptation of the well-established *multiplicative adversary method*. This new method not only unifies the compressed oracle technique within the broader adversary framework but also incorporates recent generalisations of the compressed oracle technique, providing a concrete method to extend its applicability.

Nederlandse samenvatting

Computers hebben onze manier van problemen oplossen compleet veranderd. Van de eerste rekenmachines tot moderne supercomputers: ze stellen ons in staat om steeds grotere en complexere berekeningen uit te voeren. Tegenwoordig zijn computers onmisbaar in de wetenschap, techniek en het dagelijks leven. Ze helpen ons bij het analyseren van grote hoeveelheden data, het simuleren van natuurkundige processen, en het automatiseren van ingewikkelde taken. Achter deze vooruitgang zit een belangrijk concept: algoritmen—duidelijke stap-voor-stap instructies die computers gebruiken om een probleem efficiënt op te lossen.

Hoewel het woord 'algoritme' misschien technisch klinkt, gebruiken we dit concept in feite elke dag zonder erbij na te denken. Een recept is een goed voorbeeld: het vertelt je welke ingrediënten je nodig hebt, hoe je ze moet combineren en in welke volgorde je ze moet bereiden. Dit lijkt sterk op hoe een computer een taak uitvoert met een algoritme. Veel basisprincipes van programmeren kun je herkennen in koken: Conditionele acties lijken op het nemen van beslissingen in recepten (*"Als de saus te zuur is, voeg dan suiker toe*"), iteratieve processen komen overeen met herhaalde taken (*"Roer voortdurend tot de saus is geëmulgeerd*"), en parallellisme komt overeen met multitasking (*"Terwijl de aubergine braadt in de oven, maak je de saus klaar*").

In tegenstelling tot klassieke computers, die werken volgens de wetten van de klassieke mechanica, maken kwantumcomputers gebruik van principes uit de kwantummechanica, zoals superpositie en verstrengeling. Dankzij deze principes kunnen kwantumcomputers sommige (maar zeker niet alle) berekeningen veel sneller uitvoeren dan klassieke computers. Dit geldt bijvoorbeeld voor het ontbinden van grote getallen in priemfactoren, het zoeken in ongestructureerde databases, en het simuleren van natuurkundige systemen. Binnen het veld van kwantumcomputatie onderzoeken wetenschappers hoe krachtig kwantumalgoritmen zijn en waar hun grenzen liggen. Ze proberen te begrijpen voor welke problemen kwantumcomputers echt sneller zijn en voor welke problemen ze geen voordelen bieden. Dit helpt bij het bepalen waar kwantumtechnologie in de toekomst het meest nuttig kan zijn.

Hoewel kwantumcomputers indrukwekkende voordelen bieden, is het ontwerpen van kwantumalgoritmen een grote uitdaging. Klassieke algoritmen volgen vaak een logische, intuïtieve structuur, maar kwantumalgoritmen vereisen een compleet andere manier van denken. Het proces lijkt minder op het volgen van een recept en meer op het componeren van muziek. Kwantumcomputers gebruiken interferentie om juiste oplossingen te versterken en verkeerde oplossingen uit te doven. Dit werkt net als muzikale harmonie: constructieve interferentie zorgt voor een mooi akkoord, terwijl destructieve interferentie valse tonen wegneemt. Helaas kan een kleine fout in het ontwerp van de kwantumalgoritme deze delicate balans verstoren waardoor de berekening niet meer werkt. Zoals een muzikale harmonie die een beetje uit de toon valt schril klinkt, zo wordt een kwantumalgoritme dat niet perfect is afgestemd nutteloos.

De reden dat kwantumalgoritmen minder intuïtief zijn, ligt in de unieke beginselen van de kwantummechanica. In tegenstelling tot klassieke berekeningen moet elke stap van de

kwantumalgoritme omkeerbaar zijn; informatie kan niet zomaar worden gekopieerd of verwijderd. Bovendien zijn de uitkomsten van een kwantumalgoritme probabilistisch, wat betekent dat het resultaat niet altijd met volledige zekerheid vaststaat. Deze eigenschappen maken het ontwerpen van kwantumalgoritmen extra uitdagend en vereist het begrijpen van kwantumalgoritmen ook een diepgaande kennis van de kwantummechanica. In dit proefschrift onderzoeken we methoden die het begrijpen van kwantumalgoritmen eenvoudiger maken. Deze methoden vereenvoudigen de onderliggende kwantummechanische complexiteit door problemen terug te brengen tot een vorm die dichter bij klassieke berekeningen ligt. Opmerkelijk genoeg laten we zien dat deze vereenvoudiging de kracht van kwantumalgoritmen grotendeels behoudt. Dit helpt ons beter te begrijpen waar kwantumcomputers écht voordeel bieden en waar hun beperkingen liggen.

Het ontwerpen van krachtigere kwantum toevalsbewegingen

In het eerste deel van dit proefschrift onderzoeken we een specifieke strategie voor het ontwerpen van kwantumalgoritmen, bekend als kwantum toevalsbewegingen. Dit zijn kwantumversie van toevalsbewegingen (Engels: *random walks*), die vaak als basis dienen voor efficiënte zoek- en optimalisatie-algoritmen. Een toevalsbeweging is een proces waarbij iemand willekeurige stappen neemt, zoals een dronken persoon die zonder plan door een stad zijn weg naar naar huis zoekt. Elke volgende stap alleen afhangt van de huidige positie en daarom kan dit process makkelijk wiskundig worden geanalyseerd. Hierom wordt het veel gebruikt om efficiënte zoekalgoritmes te ontwikkelen.

In een van de bekendere technien om kwantum toevalsbeweging te maken, wordt er inspiratie getrokken uit hoe elektriciteit zich verspreid door een elektrisch netwerk. Hoewel deze techniek erg toegankelijk is, zelfs voor onderzoekers zonder uitgebreide kwantumkennis, heeft het een beperking: het kan hoogstens een kwadratische versnelling bieden ten opzichte van een klassieke toevalsbeweging. Dit betekent dat als een klassieke toevalsbeweging één miljoen stappen nodig heeft, de bijbehorende kwantum toevalsbeweging slechts duizend stappen nodig heeft, want $\sqrt{1.000.000} = 1.000$. In dit proefschrift zoeken we naar manieren om deze techniek te verbeteren en mogelijk een exponentiële snelheidswinst te behalen, terwijl de oorspronkelijke intuïtie behouden blijft. Met andere woorden: kunnen we de efficiëntie van kwantum toevalsbewegingen vergroten zonder de basisprincipes van toevalsbewegingen en elektrische netwerken los te laten? In dit proefschrift laten we zien dat het antwoord hierop ja is.

Eerst introduceren we in Hoofdstuk 4 het multidimensionale kwantum toevalsbewegingen raamwerk. Dit raamwerk maakt een nieuwe klasse van kwantum toevalsbewegingen mogelijk, die wél een exponentiële versnelling kunnen behalen. Dit gaat echter wel ten koste van directe analogie met elektrische netwerken. Daarom generaliseren we in Hoofdstuk 6 het concept van elektrische netwerken, zodat deze ook geschikt zijn voor multidimensionale kwantum toevalsbewegingen. Hierdoor kunnen we de sterkte van kwantum toevalsbewegingen behouden én de intuïtieve link met elektrische netwerken herstellen.

Een belangrijke toepassing van onze nieuwe kwantum toevalsbewegingen is een sneller algoritme voor het volgende probleem: we krijgen een lijst met n getallen en we moeten bepalen of er ergens k identieke getallen in voorkomen. Neem als voorbeeld $n = 100$ en $k = 2$. Voor klassieke algoritmen is dit probleem niet erg efficiënt op te lossen. In het slechtste geval moeten we bijna de hele lijst doorzoeken: als we 99 getallen hebben bekeken, kunnen we nog steeds niet uitsluiten dat het laatste getal een dubbeltelling oplevert. Dit betekent dat klassieke methodes bijna altijd een groot deel van de lijst moeten doorzoeken. Kwantumalgoritmen kunnen dit echter veel efficiënter aanpakken en onze aanpak is te zien in Hoofdstuk 5.

Beperkingen van kwantumalgoritmen

Het ontwerpen van snellere kwantumalgoritmen om rekenproblemen efficiënter op te lossen is een interessant en waardevol onderzoeksgebied. Het is echter net zo nuttig om de grenzen van kwantumalgoritmen in de kaart te brengen. Niet alleen is de beschikbaarheid van kwantumhardware van belang, maar vooral ook de tijd van onderzoekers. Dit wordt gedaan door wiskundig te bewijzen dat de uitvoertijd van elk kwantumalgoritme dat een bepaald probleem oplost niet lager kan zijn dan een bepaalde ondergrens. Met andere woorden, we tonen aan dat geen enkel kwantumalgoritme het probleem sneller kan oplossen dan deze limiet toestaat. Hoewel deze zogenaamde no-go resultaten misschien minder spectaculair lijken, spelen ze een cruciale rol: ze helpen vast te stellen welke snelheidswinsten wel en niet mogelijk zijn.

Voor klassieke algoritmen is het vaak intuïtief mogelijk om dit soort ondergrenzen te bepalen. Stel dat iemand wil inschatten hoeveel tijd nodig is om dit proefschrift te begrijpen. Een simpele aanpak zou zijn om te berekenen hoeveel tijd nodig is om een enkele pagina volledig te begrijpen, en dit te vermenigvuldigen met het aantal pagina's. Dit geeft een ruwe ondergrens, maar het is niet perfect: sommige pagina's zijn moeilijker en kosten meer tijd dan de gemakkelijkste pagina. Ditzelfde principe wordt gebruikt bij het analyseren van de complexiteit van algoritmen.

In de kwantumwereld werkt het bepalen van ondergrenzen anders dan in de klassieke context. Een "kwantumlezer" zou in theorie meerdere pagina's tegelijk kunnen lezen in superpositie of de inhoud van een stelling verstrengelen met de uitleg ervan. Hierdoor zou informatie efficiënter verwerkt kunnen worden dan klassiek mogelijk is. Dit weerspiegelt een fundamentele balans tussen kracht en intuïtiviteit: kwantumcomputers kunnen bepaalde taken sneller uitvoeren, maar het analyseren van hun limieten wordt hierdoor ingewikkelder. Deze balans zien we terug in de technieken die worden gebruikt om kwantum ondergrenzen te bewijzen. Hoewel sterkere technieken striktere ondergrenzen geven, zijn ze vaak wiskundig complexer en minder intuïtief.

In het tweede deel van dit proefschrift onderzoeken we technieken om kwantum ondergrenzen vast te stellen. We richten ons specifiek op de *compressed oracle* techniek, die opvalt omdat hij sterke ondergrenzen afleidt op een intuïtieve manier, maar helaas slechts in beperkte situaties toepasbaar is.

Wij introduceren een nieuwe methode, die bestaande methodes verbindt met de compressed oracle techniek, waardoor het afleiden van ondergrenzen intuïtiever en toegankelijker wordt. Daarnaast biedt deze verbinding nieuwe mogelijkheden om de compressed oracle techniek uit te breiden naar een breder scala aan problemen.

Acknowledgements

First and foremost, I would like to thank my supervisor, Stacey. I am grateful for your guidance and for giving me the freedom to explore whatever I found interesting, while always making sure the right doors were open along the way. Over our $4 + \epsilon$ years together, as both your research group and family grew, I never felt like less of a priority. I feel confident we can bring our karaoke to the next stage at the fabled QIP karaoke.

I would also like to thank my cosupervisor, *Krystal*. While we did not collaborate extensively, I am glad you were part of the final stage of my doctoral journey. Now that it's in writing, we can no longer make excuses—so go ahead and pick your favorite tree from your "fluitenbos" for our performance next year.

Special thanks to *ChrisS*, you have been an academic mentor to me ever since introducing me to the field of quantum computing ten years ago. From the summer project that helped me finish my bachelor's on time to the many letters of recommendation and countless opportunities, you have never stopped extending a helping hand.

I thank the members of my doctorate committee, *Ronald, Jo, Māris, Michael,* and *Alexander*, for taking the time to read my thesis and for your valuable feedback.

Thank you to my coauthor *Jianqiang*, for our endless zoom talks and your unwavering positivity about whatever technical problem we were stuck on. It was an absolute pleasure to work with you.

I would like to thank *Andrew, François, Frédéric, Simon, Yassine* and *Yi-Kai* for your hospitality and the memorable academic visits. Special thanks to *Frédéric* and *Simon* for always extending these invitation to my partner as well and I am glad that my academic journey continues alongside you both.

A warm thank you to my colleagues at QuSoft, CWI, and the adopted ones. *Adam, Ailsa, Akshay, Ake, Aldo, Alicja, Aljosja, Alvaro, Amira, Anna, Arie, Arjan, Carla, ChrisC, ChrisS, Daan, Davi, Dima, Doutzen, Dyon, Emiel, Erik, Filippo, Florian, Fons, Fran, Freek, Galina, Garazi, Giada, Gina, Hani, Harold, Harry, Hema, Henrik, Ido, Ila, Jana, Jelena, Jeroen, John, Jonas, Jordi, Jop, Joppe, Joran, Kareljan, KoenG, KoenL, Krystal, Laurens, Léo, Llorenç, Lorenzo, Luca, Ludo, Ludovico, Lynn, Mani, Marc, Māris, Marten, Martine, Matteo, Max, Maxim, Mehrdad, Michael, Minnie, Niels, Nicolas, Nikhil, Peter, Philip, Poojth, , Randy, Remco, René, Ronald, Salvatore, Sander, Sanne, Sarah, SebastianV, Seenivasan, Shane, Simona, Sophie, Stacey, Subha, Susanne, Syver, Tao, Tony, Ute, Vania, VictorL, VictorS, Vincent, Vlad, Wouter, Yanlin, Yaroslav* and *Zongbo*, thank you all for creating such a fantastic environment. A few special mentions: *Dima*, for our shared love of snow and raclette; *Laurens* and *René*, for being my work husbands at conferences; *Joran*, for the countless coffees, both in and outside the office; *Arjan, ChrisC, Lynn* for tropical Goa, and together with *Llorenç, Hema, Nikhil, Quinten, Subha* for Nikhil's fairy tale wedding; *Philip* for ijbok; my Spanish twins *Sisi* and *Zorro*, for your wonderful friendship.

Moreover, I want to thank my paranymph *Marten*—not only for your help in these final months but for your friendship over the past ten years and your supporting coffee sessions during this last year. When we struggled through Michael's Schur-Weyl duality

homework together, we were both convinced we wouldn't pursue a PhD. Being wrong back then turned out to be excellent preparation for being wrong for four years straight.

Fuat en *Furkan*, dank jullie wel voor de warme maaltijd en de net zo warme lach sinds die allereerste lahmacun, dertien jaar geleden. De Sebastiano heeft meer mensen bereikt dan mijn onderzoek ooit zal doen.

Dank jullie wel *Mark, Omar, Seb* en *Van God Los*, voor al jullie gezelligheid, hulp en het ongevraagde cynisme. Dank jullie wel voor de Frieslandweken, de wintersporten, en het Nashen in de donkere winterdagen. Bedankt dat velen van jullie je PhD eerder af hadden, zodat ik schaamteloos alles kon kopiëren. Een speciaal dankwoord voor *Niek*, voor het Friends-avontuur en al je hulp in het afgelopen jaar.

Mijn familie en schoonfamilie, dank jullie wel voor jullie onuitputtelijke steun, trots en constante hulp, waardoor ik de luxe had om mij volledig op mijn PhD te richten. Jullie hebben, wetend of onwetend, het "impostersyndroom" altijd zoveel mogelijk onderdrukt en de afgelopen vier jaar een heel stuk makkelijker gemaakt.

Tot slot wil ik *Chelsea* bedanken. Mijn verloofde, mijn partner, mijn beste vriend. Je brengt avontuur in tijden van rust en kalmte in tijden van stress. Net zoals je dit proefschrift van opmaak hebt voorzien, zo zorg jij voor de kleur in ons leven samen. Zonder jou was dit avontuur niet gelukt, en ik ben je dankbaar voor deze vier jaar, voor de vier jaar daarvoor, en voor alle jaren die nog komen zullen. Op naar Parijs ♡