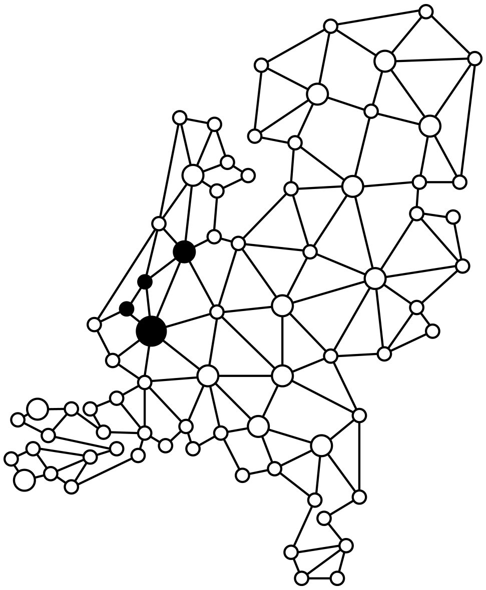
DELFT UNIVERSITY OF TECHNOLOGY

Local multi-qubit Clifford equivalence of graph states

Author
Constantijn Molengraaf



February 1, 2019

Local multi-qubit Clifford equivalence of graph states

by

Constantijn Molengraaf

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Friday February 1, 2019 at 1:00 PM.

Student number:	4217217	
Project duration:	February 12, 2018 – February 1, 2019	
Thesis committee:	Prof. dr. S.D.C. Wehner,	TU Delft, supervisor
	MSc. A. Dahlberg	TU Delft, daily supervisor
	Dr. M. Blaauboer,	TU Delft
	Dr. D. Elkouss Coronas,	TU Delft

Contents

Preface **v**

Acknowledgements **vii**

1 Introduction **1**
- 1.1 Motivation 2
- 1.2 Summary of main results 4
- 1.3 Thesis outline 6
- 1.4 Recommendations 6

2 Background **9**
- 2.1 Notation 10
- 2.2 Quantum Information 10
- 2.3 Graph Theory 14
- 2.4 Complexity theory 16

3 Review of transforming graph states **19**
- 3.1 Introducing graph states 20
- 3.2 Operations and measurements on graph states 24
- 3.3 Transforming graph states with single qubit Cliffords 29
- 3.4 Reducing stabilizer states to graph states 33
- 3.5 A quantum circuit in the graph state perspective 36

4 Introducing local multi-qubit Clifford equivalence of graph states **39**
- 4.1 Introducing T-LMQC equivalence 40
- 4.2 Brute force deciding T-LMQC equivalence 45
- 4.3 Examples of T-LMQC equivalent graphs 54

5 Mapping multi-qubit operations to ancilla states **57**
- 5.1 Introducing deciding T-LMQC equivalence via gate teleportation . . . 58
- 5.2 Formalizing the concept 60
- 5.3 The reduced Clifford set for nodes with 2 qubits 63
- 5.4 Lemmas for the proof in 5.3 72
- 5.5 Solving T-LMQC-EQUIV using the reduced Clifford set 82
- 5.6 Extension to general multi-qubit nodes 87

References **89**

A Appendix **91**

A.1 Examples of SAGE implementation 91

A.2 Generating random connected graphs 93

A.3 Stabilizer state after gate teleportation calculations 93

Preface

During the coursework of my master, I found out that I was very interested in the mathematics courses related to physics. Although I found these courses very difficult, I was very motivated, and I appreciated the beauty of the models. This also applied to the Quantum Cryptography course given by Stephanie, which really excited me. In this course, it was the first time for me that the complex mathematical models used in quantum mechanics were translated to applications in future technologies. I realised that I was very interested in learning more about this field, although I also found it very difficult. This challenged me to find a thesis project in quantum information.

Soon after I started my thesis, I realised that this would be a really challenging project. In the first few weeks, I learned about stabiliser states, graph theory and complexity theory which were new topics for me. However, I have always focussed on topics relevant for the project. This lead to the first results after about 3 months, the brute-force algorithm and the conjecture (basically chapter 4). Then, I spent some time trying to prove the conjecture. To do this, I first tried to understand Bouchet's result used in the single qubit case. The proof was only about 6 pages, so this should be do-able right? However, this work is so dense and mathematical that I never found a way to extend these results. This was an experience that I learned a lot from.

Quickly after leaving Bouchet's work, Axel came up with the idea of using gate teleportation to study the multi-qubit operations on ancilla states. Approximately the last 5 months of my thesis I have spent realizing this idea and finding out what it is useful for. This resulted in the extensive chapter 5. Working out the details of this really frustrated me, but I am delighted with the result. I learned a lot from debugging my code and my calculations, and this is definitely something that I can take with me for my next challenge.

To conclude, I am very proud of the report that you are reading right now. I hope that you learn something from it, just as I did when reading the work of my predecessors. Furthermore, I am very proud to have been part of QINC (and Qutech in general) for the last 12 months.

Acknowledgements

Pursuing a degree in Applied Physics with a specialisation in quantum technology is an enormous challenge. Here I would like to thank everyone who helped me along the way. There are a few people who deserve special attention, and I will mention them below in no particular order.

First of all, my mentor Axel. Thank you for creating such an exciting master project, and supporting me in contributing to this field. I have always very much valued our meetings, where you helped me when I was stuck, cleared my confusion about something I thought I knew, spark new ideas and give me feedback on my work. I would also like to thank Stephanie, for giving me the opportunity to work in this group. I am also grateful to all other people in the Wehner group for welcoming me and helping me on the way. Leon, thank you for joining me to YQIS. Tim, thank you for giving me some perspective when I needed it. Victoria, thank you for designing the QINC logo on the front page of my thesis.

Calculations regarding graphs can be computationally heavy. Luckily, Robbert Eggermont helped me with the problems I faced when doing calculations on the computing cluster of the Intelligent System (INSY) group at the EEMCS faculty. Thank you, Robbert!

During my thesis, I spent most of my time working (sitting) in one of the master rooms. I made a lot of new friends there, and I would like to thank everyone who shared a room with me during my thesis. However, I can not continue without mentioning a few of you personally. Mark, thank you for being there since the first hour of my study in Delft. Jarn, brother from another theory-mother, thank you for the fruitful discussions and for not eating carrots in the office anymore. Remon, thank you for teaching me something about experiments and for showing me your commutation relation document. And of course, thanks to Aletta, Bart, Carlijn, Guus, Marianne, Michael, Patrick, Romy and Sjoerd, for arranging ”broodje Leo” every Wednesday.

Last but not least, I would like to thank my family. Although my research has been (a bit) hard to understand, I always felt support and curiosity for what I was doing. Jan-Willem, thanks for thinking so much about the world and society of the future, this really inspires me. Miranda, thank you for always being there for me and remembering me to laugh every single day.

1

Introduction

The goal of this chapter is to motivate, introduce and summarize the research done in this thesis. And finally, also present a few remaining research questions in this field. Section 1.1 starts by introducing quantum internet research from a society perspective before introducing the research question of this thesis. In section 1.2 it is summarized how this work contributes to the research question. In the third section of this chapter the outline of this thesis is discussed. Finally, in section 1.4 we shortly discuss a few open questions related to the topic of this thesis. Hopefully, this chapter sparks the interest in the subject of this thesis and provides directions for where to find more on the specific results.

1.1. Motivation

Computers and the internet as we know it today have drastically changed the way we live, the way we communicate and the way we interact with the world around us. However, there are still things that the current computers and internet can not do. Therefore, new computing technologies exploiting the laws of quantum mechanics are actively being investigated in order to open a range of new applications. Such a computer - a quantum computer - promises to solve certain problems much faster than the best super computers to date. A network of (small) quantum devices, a quantum internet, has its own range of applications [1]. The most prominent is quantum key distribution (QKD), which enables fundamentally secure communication [2]. Other applications include secure access to a remote quantum computer [3] and improving observations of distant telescopes [4]. Two key phenomena of quantum mechanics which make these applications possible are superposition and entanglement. The superposition principle tells us that a binary system can not only be in one of the two states of the system, but also in a superposition of the two. Quantum systems can also be entangled, meaning that a description of one system also involves describing the other systems.

Currently, there is a race towards realizing the first scalable fault tolerant quantum computer. Even though a quantum computer outperforms a classical computer in only a few areas, the impact on society would be huge if suddenly a large quantum computer would appear today. For example, by using Shor's algorithm [5] one could break the encryption we (mostly) use nowadays for our digital communication and bank transactions. However, the encryption used in QKD can not be broken by quantum computers. Therefore, when preparing for the era of quantum computers a quantum internet would be very useful.

Often, quantum protocols ask for a specific input state before starting the actual calculations. A challenge is to determine how this specific input state, the target state, is generated. For a quantum internet, the target state could be a highly entangled state shared between multiple nodes in the network. Then, a network protocol consumes the target state in order to achieve the goal of the protocol. In general there are two ways to generate a target state. The first approach is very straight forward, the target state is generated after the protocol requests the target state. However, generating remote entanglement is a process which is costly, both in resources and in time, and quantum memories have relatively short life times. Therefore, part of the state might be lost before the target state is reached and the protocol is executed.

The second approach is based on the network generating entanglement in the background, independent of a specific target state. When a protocol requests the target state, the network already has some shared entanglement which we call the source state. When the target state is requested, the source state is then transformed to the target state by doing local operations. Local operations are typically easier to realize and have a smaller error rate compared to non-local operations, especially in a network setting. However, it is not always possible to transform the source state to the target state by doing local operations. To determine if this transformation is possible, an entanglement routing protocol is needed. The goal of this protocol is to decide which operations are needed to perform the transformation, if it is possible. When the transformation is non-destructive, i.e. the source state can be transformed to the target state and vice versa, we will call the source state and target state equivalent. Furthermore, the protocol should ideally have the following properties:

- Property 1 (General); The protocol should work for any target state demanded by the user.
- Property 2 (Compatible): The protocol should be independent of the underlying hardware.
- Property 3 (Efficient): As quantum states are short lived the protocol should be completed quickly, otherwise the quantum state will be lost.

These properties are not easy to satisfy. One obstacle is that the set of all quantum states grows very rapidly in the number of qubits. This is one of the features making quantum computing interesting. However, when trying to study properties of quantum states this rapidly growing set is not convenient. Therefore it is common practice in theoretical research to use a (smaller) set of states with an acceptable scaling which still describes the underlying features of all states. For entanglement routing, graph states turn out to be very good candidates as a subset of states. Graph states are quantum states described by simple graphs. Therefore, in order to describe a graph state on N qubits only $\frac{N(N-1)}{2}$ bits are needed, compared to the 2^N complex numbers for a general quantum state. For example, this greatly reduces the traffic in a quantum network from communication about the target state. Although the set of graph states is strictly smaller than the set of quantum states, the range of applications is not heavily reduced by only considering graph states. This is because most quantum internet protocols have graph states as an input state, for example in quantum secret sharing [6]. Furthermore, a subset of graph states are a resource for universal measurement based quantum computation (MBQC) [7]. As MBQC is able to do universal quantum computations, any quantum protocol can be executed via MBQC. Therefore, when considering only graph states for entanglement routing protocols, property 1 described above is still satisfied although not all quantum states are considered. As any stabilizer state can be transformed by single qubit operations to some graph state [8], the entanglement routing protocol for graph states can also be used for the more general problem of deciding whether a stabilizer state can be transformed to another stabilizer state. Note that for a stabilizer state on N qubits, the corresponding graph state and the necessary operations to reach this graph state can be found in $\mathcal{O}(N^3)$. Lastly, an entanglement routing protocol based on graph states is not hardware dependent. Therefore, property 2 of ideal entanglement routing protocols is satisfied.

As we restrict ourselves to graph states, the goal of the protocol reduces to whether a source graph state can be transformed to a target graph state by only doing local operations. We will only consider local operations in the Clifford group, as the action of single qubit Clifford operations mapping graph states to graph states is known in terms of the underlying graph [8]. Therefore, instead of considering quantum operations (unitary operations) on graph states, we can consider graph operations on graphs. A long standing conjecture was that any two stabilizer states equivalent under unitary operations were also equivalent under Clifford operations. However, a counter example has been provided recently [9]. It has been shown that the problem of deciding whether a source graph state can be transformed by single qubit Clifford operations, measurements and classical communication is $\mathbb{NP}$-complete [10]. All these results are obtained in the regime of one-qubit-per-node networks. The goal of this thesis is to investigate the problem of transforming graph states when local multi-qubit Clifford operations are allowed. In other words, what changes when multi-qubit Clifford operations are allowed between some nodes in the network. This problem has, to our knowledge, not been studied before.

1.2. Summary of main results

In this thesis we analyse the problem whether one graph state ($|G\rangle$, source state) can be transformed to another graph state ($|G'\rangle$, target state) by only doing local multi-qubit Clifford (LMQC) operations, where local refers to qubits inside the same node. This is formalized in definition 1.1. A graph state where some qubits share the same node is visualized in figure 1.1.

Definition 1.1 (T-LMQC-equivalence)**.** *Given two graph states $|G\rangle$, $|G'\rangle$ with the same vertex set V and a partition T of V. Two graph states are T local multi-qubit Clifford (T-LMQC) equivalent if and only if there exists a sequence of local Clifford operations which transforms $|G\rangle$ to $|G'\rangle$. T-LMQC equivalent graph states are denoted as:*

$$|G\rangle \sim_{T-\text{LMQC}} |G'\rangle \tag{1.1}$$

The problem of deciding whether two graph states are T-LMQC-equivalent is called T-LMQC-EQUIV. When T consist only of single qubit nodes, this problem reduces to the single qubit Clifford equivalence problem (SQC-EQUIV). It is known that SQC-EQUIV can be solved in $\mathcal{O}(N^4)$ [8]. On the other side, if all qubits are in the same node, the problem can be solved in $\mathcal{O}(1)$ as every source state can be transformed to any graph state on the same vertex set. The main results of this thesis are summarized below.

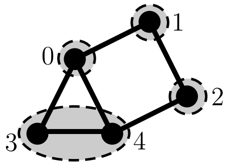
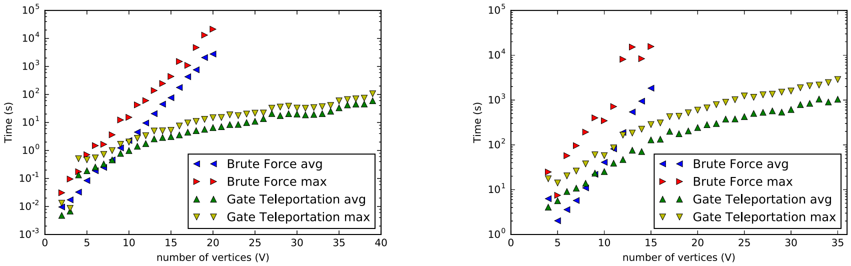


Figure 1.1: A graph $G = (V, E)$ where $V = \{0, 1, 2, 3, 4\}$ and $E = \{(0, 1), (0, 3), (0, 4), (1, 2), (2, 4), (3, 4)\}$ with partition $T = (\{0\}, \{1\}, \{2\}, \{3, 4\})$.

1. We provide an algorithm (algorithm 2) that solves T-LMQC-EQUIV. This algorithm is referred to as ”brute-force”. The running time of this algorithm scales exponentially in the size of the graph and in the number of multi-qubit nodes, but it provides a first tool to study T-LMQC-EQUIV.

2. A conjecture is presented which is inspired by the graph theory results from Bouchet [11] for the case the number of qubits per node is bounded by 2. If the conjecture is true, it can be used to construct an algorithm to solve T-LMQC-EQUIV of which the running time scales linearly in the size of the graph for a fixed number of two-qubit nodes. However, the conjecture in general does not result in an efficient algorithm. Note that there is one constraint on the input graph states if this conjecture is used. The conjecture is tested for all graph states up to 5 qubits for all allowed partitions and for all graph states on 6 qubits with 1 two-qubit node.



(a) One two-qubit node and single qubit nodes

(b) Two two-qubit nodes and single qubit nodes

Figure 1.2: Actual running times of two algorithms for the T-LMQC-EQUIV problem with a) $T = (\{0, 1\}, \dots, \{V - 1\})$ and b) $T = (\{0, 1\}, \{2, 3\}, \dots, \{V - 1\})$. ”Brute force” is algorithm 2 described in result 1., where ”Gate teleportation” is algorithm 5 mentioned in result 3. The max and average running times are calculated from at least 50 instances of the T-LMQC problem. The input graph states are random connected graphs, the partition is such that there is a) one or b) two two-qubit node(s) and all other nodes are single qubit nodes.

3. We propose another algorithm (algorithm 5) to solve T-LMQC-EQUIV based on the famous gate teleportation circuit, which we call ”Gate-teleportation”.

The algorithm is implemented and tested for instances of the T-LMQC-EQUIV problem where the number of qubits per node is bounded by 2. However, the approach is not restricted to this bound. The running time of this algorithm scales linearly in the size of the graphs but exponentially in the number of two qubit nodes. In figure 1.2 actual running times are compared for the two algorithms (”brute-force”,”Gate-teleportation”).

All algorithms described can be found on Github [12], some examples are given in appendix A.1. All results in this thesis are steps towards the ultimate goal of solving the T-LMQC-EQUIV problem in general, by either proving that it is $\mathbb{NP}$-complete or providing an polynomial-time algorithm.

1.3. Thesis outline

The results new in the thesis are presented in chapter 4 and chapter 5.

In chapter 2 we discussed background knowledge needed for the other chapters. Note that in section 2.3 graph operations (local complementations) are introduced which are fundamental for this research. Chapter 3 gives a detailed introduction to transforming graph states. No new results are presented there, except lemma 3.1. At some places we use different notations compared to literature and we have added examples. Chapter 4 introduces the T-LMQC problem and provides a first algorithm for T-LMQC-EQUIV (result 1). Furthermore, the conjecture of result 2 is discussed. Finally, some examples are discussed of T-LMQC equivalence. In chapter 5 an algorithm solving T-LMQC-EQUIV based on gate teleportation is discussed (result 3).

1.4. Recommendations

This thesis aims to be a stepping stone for other people interested in this subject, as this is the first time, to our knowledge, the T-LMQC-EQUIV problem has been studied. To stimulate further research, we will shortly discuss a few open questions.

1. Prove that T-LMQC-EQUIV is $\mathbb{NP}$-complete or that there exists a polynomial time algorithm.
2. What changes when also measurements are allowed? In other words, what is the computational complexity of deciding if a target state can be reached from a source state with LMQC operations and Pauli-measurements?
3. Related to conjecture 4.1:

(a) Prove that conjecture 4.1 is true or false. In order to do this, one probably has to extend the work of Bouchet [11] or provide a counter example.

(b) Extend conjecture 4.1 to allow for nodes with more than two qubits.

4. Related to algorithm 5 (Gate teleportation):

   (a) Does there exists a smaller reduced Clifford set than discussed in theorem 5.1?

   (b) When a target graph state can be reached from a source graph state by doing single qubit operations, local Pauli-measurements and classical communication, the target graph state is a qubit-minor of the source graph state. The corresponding decision problem is called the qubit-minor problem, which is known to be $\mathbb{NP}$-complete [10]. Is it possible to reduce the T-LMQC-EQUIV problem via gate teleportation to the qubit-minor problem by adding ancilla qubits to the source graph state?

5. In this thesis we have always assumed the partition to be fixed. However, there are also interesting questions where this is not the case. For example:

   (a) Given a source state with only single qubit nodes and the freedom to choose one location in the partition where a local qubit is added. Can the target state be reached by local multi-qubit Clifford operations and Pauli-measurements when one local qubit is added to the source state?

   (b) Given a source state and the ability to do one extra CZ between any two qubits in the graph state. Can the target state be reached by doing SQC operations after possible applying one CZ gate?

6. Can the results on the computational complexity of deciding equivalence of graph states can be used in the field of code switching in quantum error correction? In a future quantum computer, it might be useful to be able to fault-tolerantly switch between codes, as every code has it's own advantages [13],[14]. This fault-tolerant mappings have a certain notion of locality. Can this notion somehow be mapped to the notion of locality in transforming graph states?

7. In this research, operations and states are assumed to be perfect. What if they are not? I.e., what happens when the source state and/or the operations are noisy?

2

Background

This chapter covers important prerequisite knowledge for the other chapters in this thesis. The content of this chapter is usually covered in textbooks on quantum information, graph theory or computer science. Thus, we will discuss most topics only briefly and provide references for the relevant background material. However, in the section on graph theory, one important graph operation is discussed which is fundamental to later chapters. The first section introduces notations used in this thesis. The other three sections in this chapter can be read separately, and the reader could only read the ones he/she is not familiar with.

2.1. Notation

First, we introduce the general notation used throughout this thesis. We use $\mathbb{C}$ to denote the set of all complex numbers. The set $\{0, 1\}$ with addition and multiplication as in table 2.1 is the finite field of dimension 2, which is denoted by $\mathbb{F}_2$. Addition in $\mathbb{F}_2$ corresponds to addition $\mod 2$. Multiplication corresponds directly to the usual multiplication rules. There are other notations used in literature, for example $GF(2)$. The set of $N \times N$ matrices with elements in $\mathbb{F}_2$ is denoted by $\mathbb{F}_2^{N\times N}$. A vector v of dimension N with elements in $\mathbb{F}_2$ is denoted by $v \in \mathbb{F}_2^N$. $\mathcal{U}(N)$ denotes the $N \times N$ unitary group, the group of unitary matrices with matrix multiplication as group operation. Matrices will usually be written in bold.

+	0	1
0	0	1
1	1	0

(a) Addition in $\mathbb{F}_2$.

×	0	1
0	0	0
1	0	1

(b) Multiplication in $\mathbb{F}_2$.

Table 2.1: Addition and multiplication in $\mathbb{F}_2$.

The direct sum ($\oplus$) of two matrices $\mathbf{A}$ and $\mathbf{B}$ of dimensions $m \times n$ and $p \times q$ respectively, is defined as in equation 2.1. The tensor product ($\otimes$) of two vertices v, w with dimensions n, m is defined in equation 2.2.

$$\mathbf{A} \oplus \mathbf{B} = \begin{bmatrix} \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \cdots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} & \mathbf{0} \\ \mathbf{0} & \begin{bmatrix} b_{11} & \cdots & b_{1q} \\ \vdots & \cdots & \vdots \\ b_{p1} & \cdots & b_{pq} \end{bmatrix} \end{bmatrix} \tag{2.1}$$

$$v \otimes w = \begin{bmatrix} v_1 w_1 & v_1 w_2 & \cdots & v_1 w_m \\ v_2 w_1 & v_2 w_2 & \cdots & v_2 w_m \\ \vdots & \vdots & \ddots & \vdots \\ v_n w_1 & v_n w_2 & \cdots & v_n w_m \end{bmatrix} \tag{2.2}$$

2.2. Quantum Information

As we will be dealing with quantum states a lot, we will first introduce the notation used for quantum states. The basis of a classical computer is a classical bit, a zero or an one. In a quantum computer, quantum bits or qubits are used which can be either zero, one or a superposition of zero and one. The zero state is denoted as

$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and the one state as $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. A (pure) quantum state of one qubit, $|\psi\rangle$, is given by $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ where $|\alpha|^2 + |\beta|^2 = 1$ and $\alpha, \beta \in \mathbb{C}$. This state can be a superposition of $|0\rangle$ and $|1\rangle$ depending on the choice of α and β. It is possible to use a different basis and a commonly used one is the plus minus basis: $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. A (pure) quantum state of N qubits can be described with the same approach, where $2^N - 1$ pre-factors determine the state. The set of matrices $\{\mathbb{1}, X, Y, Z\}$ is called the set of Pauli matrices where:

$$\mathbb{1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{2.3}$$

One can see that the vectors $|0\rangle$ and $|1\rangle$ are actually the eigenvectors of Z, just like $|+\rangle$ and $|-\rangle$ are the eigenvectors for X. The H (Hadamard) gate, S (phase) gate are defined as follows:

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \tag{2.4}$$

Next to single-qubit gates, we also introduce two-qubit gates. We will mostly use the controlled-Z (CZ) and controlled-X ($CNOT$) gate. A subscript is used to denote on which qubit(s) the gate is applied. For i, j as the control and target qubit respectively, we have:

$$CZ_{i,j} = |0\rangle\langle 0|_i \otimes \mathbb{1}_j + |1\rangle\langle 1|_i \otimes Z_j, \quad CZ_{0,1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{2.5}$$

$$CNOT_{i,j} = |0\rangle\langle 0|_i \otimes \mathbb{1}_j + |1\rangle\langle 1|_i \otimes X_j, \quad CNOT_{0,1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.6}$$

A few relations between the gates introduced above are given in equation 2.7.

$$\begin{gathered} X = iZY, \quad Y = iXZ, \quad Z = iYX \\ HXH = Z, \quad HZH = X, \quad HYH = -Y \\ (\mathbb{1} \otimes Z)CZ = CZ(\mathbb{1} \otimes Z), \quad (\mathbb{1} \otimes X)CZ = CZ(Z \otimes X) \end{gathered} \tag{2.7}$$

Using these equations one can easily show that two different Pauli's always anti-commute, i.e. for $P_1, P_2 \in \{X, Y, Z\}$ and $P_1 \neq P_2$, then $P_1P_2 + P_2P_1 = \{P_1, P_2\} = 0$. When a qubit is measured it is projected onto a certain basis. The outcome will be either a 0 or a 1, corresponding to the $+1$ or -1 eigenvalue respectively. After the measurement, the state is collapsed to the eigenvector corresponding to the measurement outcome. We usually consider measurements in the Pauli basis X, Y

or Z. This corresponds to projecting the measured state to one of the eigenvectors of the Pauli matrix corresponding to the chosen basis.

Often we only need a subset of all elements of a group to describe all elements in the group. So a subset $g_1, ..., g_l$ (the generating set) for $g_1, ..., g_l \in G$ can be used to construct every element in G (the generated group) under the group operation. We write this as $G = \langle g_1, ..., g_l \rangle$. Note that a generating set is not necessarily unique. The one qubit Pauli group ($\mathcal{P}^1$) is the group with all Pauli matrices and a multiplicative factor $\pm 1, \pm i$: $\mathcal{P}^1 = \{\pm\mathbb{1}, \pm i\mathbb{1}, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}$. Sometimes the group is defined without the imaginary unit phase, but note that this is not a group: $\mathcal{P}^1_\pm = \{\pm\mathbb{1}, \pm X, \pm Y, \pm Z\}$. A generating set for the Pauli group is given by $\langle X, Z, i\mathbb{1}\rangle$. The Pauli group for N qubits, $\mathcal{P}^N$, is obtained by taking N times the tensor product with the single qubit Pauli group: $\mathcal{P}^N = (\mathcal{P}^1)^{\otimes N}$. As we will be dealing with large number of qubits it is convenient to introduce some compact notations. To denote a N-qubit state where all qubits are in the state $|+\rangle$, we use:

$$|+\rangle^N = \otimes_{i=1}^N |+\rangle^i \tag{2.8}$$

The same notation is also used for operators and sets. One example of such a multi-qubit state is the Greenberger Horne Zeilinger state (GHZ-state):

$$|GHZ\rangle_N = \frac{\otimes_{i=1}^N |0\rangle^i + \otimes_{i=1}^N |1\rangle^i}{\sqrt{2}} = \frac{|0\rangle^N + |1\rangle^N}{\sqrt{2}} \tag{2.9}$$

There is a special name for the group of unitary operations which leaves the Pauli group invariant under conjugation. This group is called the Clifford group $\mathcal{C}^N$ where N refers to the number of qubits considered:

$$\mathcal{C}^N = \{U \in \mathcal{U}(2^N) : (\forall P \in \mathcal{P}^N : UPU^\dagger \in \mathcal{P}^N)\} \tag{2.10}$$

When we restrict ourselves to operations on one qubit, this reduces to the single qubit Clifford group:

$$\mathcal{C}_1^N = \{U \in \mathcal{U}(2)^N : (\forall P \in \mathcal{P}^N : UPU^\dagger \in \mathcal{P}^N)\} \tag{2.11}$$

One of the reasons the Clifford group is interesting is that quantum circuits with only Clifford operations, Pauli measurements and computational basis initialization can be simulated efficiently on classical computers [15]. Furthermore, the Clifford group together with a gate not in the Clifford group forms an universal gate set. The set $\{H, S, CZ\}$ generates the multi-qubit Clifford group [15]. The single qubit Clifford group is generated by $\{H, S\}$ [16].

The size of the Clifford group is not directly clear from the definition. However, the size of the Clifford is very relevant for the later stages of this thesis. Here we will discuss the size of the N-qubit Clifford group [17]. Let $C \in \mathcal{C}^N$. We will consider the mapping of a Pauli by the Clifford C. First, we note that identity must be mapped to itself, which follows from:

$$\mathbb{1}_N \mapsto C\mathbb{1}_N C^\dagger = CC^\dagger \mathbb{1}_N = \mathbb{1}_N \tag{2.12}$$

Let $P, P' \in \{X, Y, Z\}$. Then it follows that P can not map to iP', because then $\mathbb{1} = PP \mapsto CPC^\dagger CPC^\dagger = iP'iP' = -\mathbb{1}$. Furthermore, note that if $CPC^\dagger$ is known, $CiPC^\dagger$ is known as a scalar commutes with all matrices. Next, we will need all non-identity Pauli matrices:

$$\bar{P}^N = \{\sigma_1 \otimes \cdots \otimes \sigma_N | \sigma_i \in \{\mathbb{1}, X, Y, Z\}\} \backslash \{\mathbb{1}^{\otimes N}\} \tag{2.13}$$

Then from above reasoning, it follows that it is sufficient to specify how C transforms $\bar{P}^N$ in order to know how $P^N \in \mathcal{P}^N$ is transformed. Using the fact that $XZ = iY$ and $XX = ZZ = \mathbb{1}$, it is sufficient to only specify the X and Z part to know how Y is transformed. Let us consider $C \in \mathcal{C}^N$ and $P \in \bar{P}_N$. We will start from the last pair (X_n, Z_n), i.e. the X and Z part of the nth qubit. X_n can be mapped to any element of $\pm\bar{P}^N$. Therefore, there are $2\left|\bar{P}^N\right| = 2(4^N - 1)$ choices for X_n. The Z_n element can be mapped to any element of $\pm\bar{P}^N$ that anti-commutes with $CX_nC^\dagger$, which gives $\frac{1}{2} 2 \times 4^N = 4^N$ possibilities. The same then goes for the $N-1$th element of P, which finally results in the following expression for the size of the N-qubit Clifford group:

$$|\mathcal{C}_N| = \prod_{i=1}^{N} (2 \times 4^i - 2)(4^i) \tag{2.14}$$

Let us consider this result for the 1-qubit Clifford group. For the 1-qubit group, we have $\bar{P}^1 = \{X, Y, Z\}$ and let $C \in \mathcal{C}_1$. Therefore, X can be mapped to any element of $\pm\bar{P}^1 = \{\pm X, \pm Y, \pm Z\}$, which are 6 possibilities. Then Z can be transformed to any element of $\pm\bar{P}^1 \backslash \{\pm CXC^\dagger\}$, which gives 4 possibilities. Therefore, $|\mathcal{C}_1| = 6 \times 4 = 24$.

Note that in general the size of the Clifford group scales exponentially. For N up to 4 the size of the Clifford group is given in table 2.2.

N	$\lvert\mathcal{C}_N\rvert$
1	24
2	11520
3	92897280
4	12128668876800

Table 2.2: The size of the Clifford group on N qubits for N up to 4.

One way to represent quantum computations is to use quantum circuits. Lines represent qubits and rectangles or gates represent unitary operations, a square with a meter represents computational basis measurements and a double line represents a classical bit. Figure 2.1 is an example of such a circuit.

For an increasing number of qubits it can be complicated and inefficient to keep track of all the coefficients. For some pure states it is possible to use a different description, the stabilizer formalism. In this stabilizer formalism a quantum state

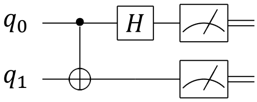


Figure 2.1: An example of a quantum circuit of two qubits (q_0, q_1). This circuit implements the well known Bell measurements. First a CNOT is applied with q_0 as control qubit and q_1 as target qubit. Afterwards a Hadamard (H) gate is applied on qubit q_0. Finally, both qubits are measured in the Z/computational/standard basis. The outcome of the circuit is two classical bits (double lines).

is not described by its coefficients, but it is described by a set of operators which map the state to itself. We say that the state is stabilized by this set of operations.

Let $\mathcal{S}$ be a set of N independent commuting elements from $\mathcal{P}^N$ not containing $-\mathbb{1}^N$. The N-qubit quantum state corresponding to the vector with eigenvalue $+1$ for all elements in $\mathcal{S}$ is called the stabilizer state. Furthermore, $\mathcal{S}$ is called the stabilizer of the stabilizer state.

One can directly see that all elements in $\mathcal{S}$ have to commute, if the elements would anti-commute the only vector stabilized by these elements would be the zero vector. The minimum number of independent elements needed to generate $\mathcal{S}$ is called the rank (r_S) of the stabilizer.

$$r_S := \text{rank}(S) = \min\{n|\langle\{s_1, ..., s_n\}\rangle = S, s_i \in S\} \tag{2.15}$$

Every $r_S = N$ generating set stabilizes exactly one quantum pure state up to a phase factor, where N is the number of qubits.

2.3. Graph Theory

In this section we will introduce graphs and discuss properties of graphs which turn out to be useful later in this research. Let's start with defining a graph.

A graph G is a tuple of a set of vertices V and a set of edges E. A vertex $v \in V$ can be visualized as a node in a network, where an edge $e \in E$ connects two vertices in this network. An edge e is denoted as a starting vertex u and a receiving vertex v: $e = (u, v)$. When the direction of the edge does not matter we have an undirected graph, i.e. edges are then 2-element subsets of V. A self connecting edge is an edge with the same start vertex as end vertex: $e = (u, u)$. A graph where the number of edges between two vertices is either zero or one, is a graph without multiple edges. By defining E as a set, we already restricted ourselves to graphs without multiple edges. A graph without multiple edges and without self connecting edges is called a simple graph. An example is given in figure 2.2. In this thesis we will only consider simple graphs.

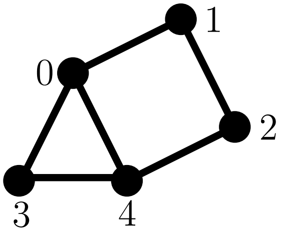


Figure 2.2: A graph $G = (V, E)$ where $V = \{0, 1, 2, 3, 4\}$ and $E = \{(0, 1), (0, 3), (0, 4), (1, 2), (2, 4), (3, 4)\}$.

When there is an edge between two vertices we call the vertices adjacent. The adjacency matrix Γ of a graph G is defined as:

$$\Gamma_{u,v} = \begin{cases} w_{u,v} & \text{if } (u,v) \in E \\ 0 & \text{otherwise} \end{cases} \tag{2.16}$$

Where $w_{u,v}$ is the weight of the edge (u, v). For an unweighted graph all edges have weight 1 ($w_{u,v} = 1 \ \forall \ (u, v) \in E$). The neighbourhood N_u of a vertex u is the set of vertices v which are adjacent to u. In other words: $N_u := \{v \in V | u, v \in E\}$. The size of N_u ($|N_u|$) is called the degree of u. A vertex with degree zero, i.e. $|N_u| = 0$, is called an isolated vertex. Note that for simple graphs the degree of a vertex is upper bounded by $|V| - 1$. A connected graph is a graph where every vertex is reachable from another vertex by moving only between adjacent vertices. We use the notation $G \backslash u$ for the resulting graph after vertex u has been removed from G. This notation is also used for removing a set of edges F from G: $G \backslash F$. For two sets (of vertices for example) V,W the union of V and W is denoted by $V \cup W$. For the intersection of two sets we use $V \cap W$. The symmetric difference is written as a plus sign and is defined as: $V + W = (V \cup W) \backslash (V \cap W)$. With a little bit abuse of notation, we also use the symmetric difference for flipping edges in a graph: $G + (a, b) \equiv E[G] + \{(a, b)\}$.

It could be that at some point we want to, for example, colour some vertices red and some vertices blue. It would be useful to have a way of labelling which vertices are red and which are blue. For this we introduce a partition T of V. The partition of V, $(A_1, ..., A_m)$, is a tuple where every element of the tuple, a part, is a set of one or more vertices and different parts do not overlap, i.e. $\forall i, j : A_i \cap A_j = \emptyset$ and $\bigcup_{i=1}^{m} A_i = \{0, \dots, V - 1\}$.

There is one graph operation that we will use often, which is called local complementation. In chapter 3 we will see how local complementing a graph corresponds to transforming a graph state with single qubit Clifford operations. Complementing a graph G is the operation where every edge in G is removed and every edge not in G is added. The result is the complement of G, or G^c.

Local complementations

Definition 2.1. *Given a graph G and a vertex $u \in V[G]$. Local complementing a vertex u of G refers to complementing the subgraph induced by the neighbourhood of u, N_u. The other parts of G are left unchanged. The resulting graph is denoted by $\tau_u(G)$. In terms of the adjacency matrix, local complementing G with adjacency matrix Γ_G on vertex u gives the graph with adjacency matrix $\Gamma_{\tau_u(G)}$:*

$$\Gamma_{\tau_u(G)} = \Gamma_G + \Theta(N_u) \mod 2 \tag{2.17}$$

Here $\Theta(N_u)$ refers to the complete graph's adjacency matrix on the neighbourhood N_u of u where zero elements are added to match the dimensions of Γ_G.

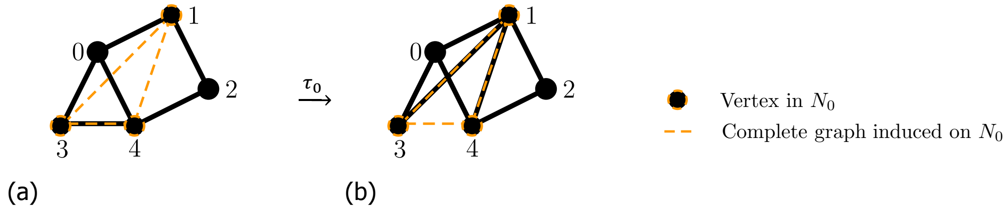


Figure 2.3: An example of local complementing a graph G at vertex 0, which we denote by τ_0 on G. The orange dashed circles represent the vertices adjacent to 0, whereas the orange dashed lines represent the complete graph induced by this neighbours.

2.4. Complexity theory

In this section we will briefly introduce the topics from computer complexity theory which we will use in this thesis. Computer complexity theory provides a formal measure of computational complexity of a problem, which makes it possible to compare the complexity of different problems and algorithms.

When thinking about the complexity of a problem, one of the first measures you might think about is the amount of time you need to spend on the problem to solve it. For example for a puzzle of 100 pieces one could ask the amount of time you will need to solve it completely. The time needed depends at least on n, the number of pieces of the puzzle. It is not directly clear in what time you have to solve the problem in order to receive the label 'fast'. For computers solving problems there is a common way to denote the complexity of a problem. Let n be the size of the problem. Deterministic algorithms which give a solution in polynomial time ($\mathcal{O}(n^{\mathcal{O}(1)})$) are considered to be fast. However, algorithms which scale exponentially, $\mathcal{O}\left(2^{\text{poly}(n)}\right)$, are considered to be slow. To conclude, polynomial time algorithms are preferred to exponential time algorithms.

When a certain algorithm runs in exponential time, it might still be useful to know in more detail how different parameters contribute to this scaling. Fixed parameter tractable problems scale exponential in one parameter m and polynomial in the size of the input to the problem n: $\mathcal{O}\left(2^{\text{poly}(m)} n^{\mathcal{O}(1)}\right)$. For small m, these problems could still be solved in reasonable time.

3

Review of transforming graph states

This chapter is written as an introduction to the field of graph states and the corresponding transformations. No new results are presented. The topics discussed here are specifically tailored to the needs of the later chapters. As most of the topics covered are not standard graduate course content, we will also include some examples or extra discussion to help the reader gain some intuition. In section 3.1 we start by introducing graph states. The effect on a graph when measuring or transforming the underlying quantum state is discussed in 3.2. In section 3.3 the problem of deciding equivalence under single qubit Clifford operations is described and an efficient algorithm is discussed. The relation between stabilizer states and graph states is discussed in section 3.4. The goal of section 3.5 is twofold. First it bundles the section 3.1 and 3.2 by discussing an example of graph state operations. Furthermore, the well known teleportation circuit is discussed in the graph state picture. This circuit will be the foundation for the results of chapter 5.

3.1. Introducing graph states

The goal of this section is to introduce graph states and the corresponding mathematical framework. In section 3.1.1 graph states are defined. Afterwards, in section 3.1.2, we introduce the binary representation of stabilizer states and graph states. This binary representation is used in section 3.1.3 to discuss equivalence of graph states and stabilizer states.

3.1.1. Introducing graph states

In this section we introduce graph states. We start by giving the definition of graph states [18].

Graph states

A graph state $|G\rangle$ is a quantum state which directly relates to a graph $G = (V, E)$. Every $v \in V$ corresponds to a qubit initialized in the state $|+\rangle$ and every edge $(u, v) \in E$ corresponds to a controlled-z (CZ) operation between qubits u and v. I.e.:

$$|G\rangle = \prod_{(u,v)\in E(G)} CZ^{(u,v)} \left(\otimes_{v\in V(G)} |+\rangle_v\right) \tag{3.1}$$

A visualization of a graph state is shown in figure 3.1. One commonly used quantum state is the GHZ-state on n qubits. The star graph (figure 3.2) and the complete graph (figure 3.3) can both be transformed to the GHZ-state by only doing single qubit Clifford operations. The complete graph on 2 qubits, as shown in figure 3.4, is equivalent to a Bell pair up to single qubit Clifford operations. Therefore, we will usually refer to this state as the Bell pair graph state. Note that by doing Hadamards on the leaves of a star graph, the GHZ-state is obtained.

3.1.2. Binary representation

In this section we will introduce the binary notation for stabilizer states. This is another way to describe the stabilizer formalism from section 2.2 which will turn out to be quite useful. The element u from the Pauli group can be described up to a global phase by a vector $\mathbf{U} = (\mathbf{U}_x, \mathbf{U}_z)$ where $\mathbf{U} \in \mathbb{F}_2^{2N}$ by:

$$u = X^{\mathbf{U}_x} Z^{\mathbf{U}_z} \equiv \prod_{a\in V} X^{U_x^a} \prod_{a\in V} Z^{U_z^a} \tag{3.2}$$

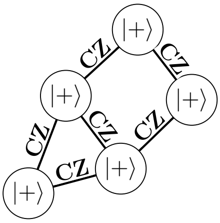


Figure 3.1: A graph state $|G\rangle$ corresponding to the graph $G = (V, E)$. Every vertex corresponds to a qubit initialized in the $|+\rangle$ state, every edge corresponds to a CZ gate between the two vertices.

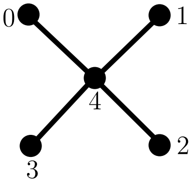


Figure 3.2: $S_{\{0,1,2,3,4\},4}$, the star graph on the vertex set $\{0, 1, 2, 3, 4\}$ with centre 4.

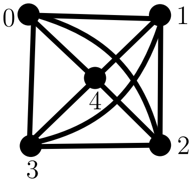


Figure 3.3: The complete graph on 5 vertices.

Figure 3.4: The complete graph on 2 vertices. This state can be transformed to a Bell pair by doing single qubit Clifford operations.

It is common to use the following notation for Pauli operators in symplectic vector form:

$$\mathbb{1} \mapsto (0|0), \quad X \mapsto (1|0), \quad Y \mapsto (1|1), \quad Z \mapsto (0|1) \tag{3.3}$$

For a string of Paulis, lets take $X^1 Z^2 \mathbb{1}_2^3 Y^4 Z^5$ for example, on different qubits we find:

$$X^1 Z^2 \mathbb{1}_2^3 Y^4 Z^5 \mapsto (10010|01011) \in \mathbb{F}_2^{2\cdot 5} \tag{3.4}$$

There are a few important properties belonging to this binary representation. Before we state these properties, we first introduce the symplectic inner product $\mathbf{P} = \left[\begin{array}{c|c} \mathbf{0}_N & \mathbb{1}_N \\ \hline \mathbb{1}_N & \mathbf{0}_N \end{array}\right]$.

Properties binary representation [18]

Letting $u, v, w \in \mathcal{P}^N$ with corresponding binary vectors $\mathbf{U}, \mathbf{V}, \mathbf{W} \in \mathbb{F}_2^{2N}$, then:

$$uv \sim w \quad \leftrightarrow \quad \mathbf{U} + \mathbf{V} = \mathbf{W} \tag{3.5a}$$

$$[u, w] = 0 \quad \leftrightarrow \quad \mathbf{U}^T\mathbf{P}\mathbf{W} = \mathbf{0} \tag{3.5b}$$

A subspace $\mathbf{S}$ of $\mathbb{F}_2^{2N}$ is called self-dual if and only if:

1. $\mathbf{U}^T\mathbf{P}\mathbf{W} = 0$ for every $\mathbf{U}, \mathbf{W} \in \mathbf{S}$.
2. if $\mathbf{W} \in \mathbb{F}_2^{2N}$ and $\mathbf{W}^T\mathbf{P}\mathbf{U} = 0$ for every $\mathbf{U} \in \mathbf{S}$, then $\mathbf{W} \in \mathbf{S}$.

There is a one-to-one relation between a stabilizer group $\mathcal{S}$ on N qubits with stabilizer state $|\mathcal{S}\rangle$ and the N-dimensional, self-dual linear subspace $\mathbf{S}$ of $\mathbb{F}_2^{2N}$ in the binary representation. Just as before we use the generators of this subspace to describe it. We do this by listing the generators in the generating set in binary form as rows of the generator matrix $(\mathbf{X}|\mathbf{Z})$. The dimensions of this generator matrix are $2N$ by N. We know, from the definition of stabilizers, that all elements of the stabilizer group should commute. From the properties of the binary representation, specifically equation 3.5, this can be stated in the binary representation as follows:

$$(\mathbf{X}|\mathbf{Z})\mathbf{P}(\mathbf{X}|\mathbf{Z})^T = 0 \tag{3.6}$$

Furthermore, there is a clear relation between the generator matrix of a graph state and its adjacency matrix:

Adjacency matrix ~ generator matrix

The generator matrix for a graph state $|G\rangle$ with adjacency matrix Γ has the following standard form:

$$(\mathbf{X}|\mathbf{Z}) = (\mathbb{1}_N|\Gamma) \tag{3.7}$$

3.1.3. Equivalence of graph states/stabilizer states

In this section we will discuss the question whether two stabilizer states are equivalent under Clifford operations. Therefore, we first introduce the symplectic vectorspace. A symplectic matrix is defined as a matrix $\mathbf{Q} \in \mathbb{F}_2^{2N\times 2N}$ for which holds that:

$$\mathbf{Q}^T\mathbf{P}\mathbf{Q} = \mathbf{P} \tag{3.8}$$

A symplectic matrix can be seen as four $N \times N$ matrices organized as two by two block matrix. It is shown [18] that every Clifford operation U corresponds to a symplectic transformation $\mathbf{Q}$. Furthermore, every symplectic transformation $\mathbf{Q}$ can be realized as a Clifford operation [19] up to a phase. Two stabilizer states $\mathcal{S}$, $\mathcal{S}'$ are equivalent under Clifford operations if and only if there exists two symplectic matrices $\mathbf{Q}$, $\mathbf{Q}'$ such that $\mathbf{Q}\mathcal{S} \mapsto \mathcal{S}'$ and $\mathbf{Q}\mathcal{S}' \mapsto \mathcal{S}$. Before we formalize the equivalence of stabilizer states, we first discuss the freedom to choose a generating set for a stabilizer.

A generating set for a stabilizer is not unique in the sense that there could be different generating sets describing the same stabilizer. To transform one generating set to another generating set of the same stabilizer, only a basis change is needed. Note that this is not the same as the symplectic transformations discussed before. The basis change can be achieved by an invertible N by N matrix $\mathbf{R}$, and more importantly a basis change is no physical operation. For two different generator matrices of the same stabilizer group, it then follows that: $(\mathbf{X}'|\mathbf{Z}') = \mathbf{R}(\mathbf{X}|\mathbf{Z})$.

Let us now formalize the equivalence of two stabilizers under Clifford operations. Using the notation introduced previously, there are three equivalent ways to do this.

Equivalence of stabilizer states [18]

Letting $\mathcal{S}$, $\mathcal{S}'$ be full rank stabilizers with generator matrices $(\mathbf{X}|\mathbf{Z})$,$(\mathbf{X}'|\mathbf{Z}')$. Then there are three equivalent statements for equivalence of stabilizer states:

$$\mathcal{S}' = U\mathcal{S}U^{\dagger} \text{ for some } U \in \mathcal{C}_N \text{ with corresponding symplectic matrix } \mathbf{Q} \tag{3.9a}$$

$$(\mathbf{X}'|\mathbf{Z}') = \mathbf{R}(\mathbf{X}|\mathbf{Z})\mathbf{Q}^T \text{ for some invertible } \mathbf{R} \tag{3.9b}$$

$$\text{and for some } \mathbf{Q} \in \mathbb{F}_2^{2N\times 2N} \text{with corresponding } U$$

$$(\mathbf{X}'|\mathbf{Z}')\mathbf{P}\mathbf{Q}(\mathbf{X}|\mathbf{Z})^T = \mathbf{0} \text{ for some } \mathbf{Q} \in \mathbb{F}_2^{2N\times 2N} \text{with corresponding } U \tag{3.9c}$$

It has been shown [8] that every stabilizer state can be transformed to a graph state by single qubit Cliffords and that the corresponding Clifford and graph state can be calculated efficiently. This means that deciding equivalence of two stabilizer states can be reduced to deciding equivalence of two graph states, because we can always transform stabilizer states to graph states. Note that every row in the generator matrix of a stabilizer has a ± 1 phase, but after transforming the stabilizer to a graph state all rows will have a $+1$ phase. Suppose we have two graph states $|G\rangle$ and $|G'\rangle$ with corresponding adjacency matrices Γ, Γ'. The remaining question is whether the two graph states are equivalent according to equation 3.5, where we will use the third formulation. If we can find a symplectic matrix $\mathbf{Q}$ such that

$(\mathbf{X}'|\mathbf{Z}')\mathbf{P}\mathbf{Q}(\mathbf{X}|\mathbf{Z})^T = \mathbf{0}$ holds, we know that $|G\rangle$ and $|G'\rangle$ are equivalent (and that the underlying $\mathcal{S}, \mathcal{S}'$ are equivalent).

We can write the symplectic matrix $\mathbf{Q}$ as follows:

$$\mathbf{Q} = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array}\right] \tag{3.10}$$

Where $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ are N by N matrices with elements in $\mathbb{F}_2$. Plugging this into equation 3.9c we find:

$$\Gamma'\mathbf{B}\Gamma + \mathbf{D}\Gamma + \Gamma'\mathbf{A} + \mathbf{C} = \mathbf{0} \tag{3.11}$$

In order to check whether two graph states are equivalent, we have to find $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ such that both equation 3.11 and equation 3.8 hold. This last constraint comes from the fact that $\mathbf{Q}$ has to be a symplectic transformation. In order to do this we can calculate the basis $\mathcal{B}$ of the set of vector space formed by the solutions $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ for equation 3.11. In order to verify if a solution exists we have to check all possible combinations of basis vectors in $\mathcal{B}$ against the constraint 3.8. As $\mathcal{B}$ in general is of dimension $\mathcal{O}(N)$, this could lead to $\mathcal{O}(2^N)$ solutions to check against the symplectic constraint. We will see later how this number can be reduced.

3.2. Operations and measurements on graph states

Here we will relate Clifford operations to graph operations. First, we relate single qubit Clifford operations to local complementations. Therefore, we use a compact notation for an operation U on a set of qubits W: $U[W] = \prod_{i \in W} U_i$.

Definition 3.1 ([18]). *Given a graph state $|G\rangle$ and let u be a vertex in G, i.e. $u \in V[G]$. U_u is given as:*

$$U_u = \sqrt{-X}[u]\sqrt{Z}[N_u] \tag{3.12}$$

Applying U_u to $|G\rangle$ corresponds to local complementing the graph G at vertex u, i.e.:

$$U_u |G\rangle = |\tau_u(G)\rangle \tag{3.13}$$

U_u can also be written as $U_u = e^{-i\frac{\pi}{4}X^u} e^{i\frac{\pi}{4}Z^{N_u}}$. Here we see how local complementing a graph corresponds to single qubit Clifford operations on a graph state. An example is given in figure 3.5. In general, every single qubit Clifford operation mapping graph states to graph states can be decomposed in a sequence of local complementations [8].

One sequence of local complementation deserves its own name, the pivot operation.

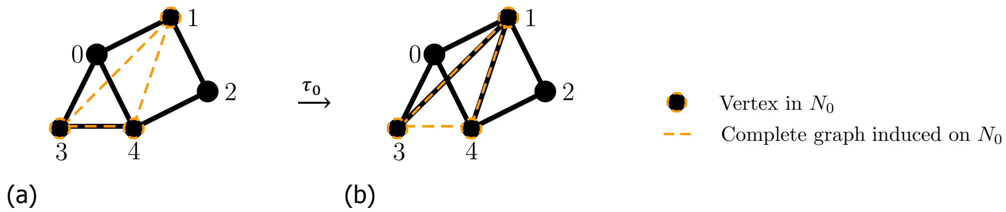


Figure 3.5: An example of local complementing $|G\rangle$ at qubit 0 which we denote by τ_0 on $|G\rangle$. The graph in figure (b) is denoted by $|\tau_0(G)\rangle$.

Definition 3.2 ([10]). *With $(u, v) \in E[G]$, the graph operation $\rho_{u,v}$ takes a graph G to $\rho_v(G)$ such that:*

$$\rho_{u,v}(G) = \tau_v \tau_u \tau_v(G) \tag{3.14}$$

We will now introduce some notation in order to see how a pivot transforms a graph state directly, i.e. not by doing local completementations. Given that $a, b, c, d \in V[G]$ are four distinct vertices. To denote the sets of neighbours in G of a and b, a but not b and a but not b as $N^G_{ab} = N_a(G) \cap N_b(G)$, $N^G_{a\backslash b} = N_a(G)\backslash(N_b(G) \cup b)$ and $N^G_{b\backslash a} = N_b(G)\backslash(N_b(G) \cup a)$.

Definition 3.3. *Two vertices c and d are from a different neighbourhood of (a, b) if $c \in N^G_i$ and $d \in N^G_j$ for $i \neq j$ and $i, j \in \{ab, a\backslash b, b\backslash a\}$.*

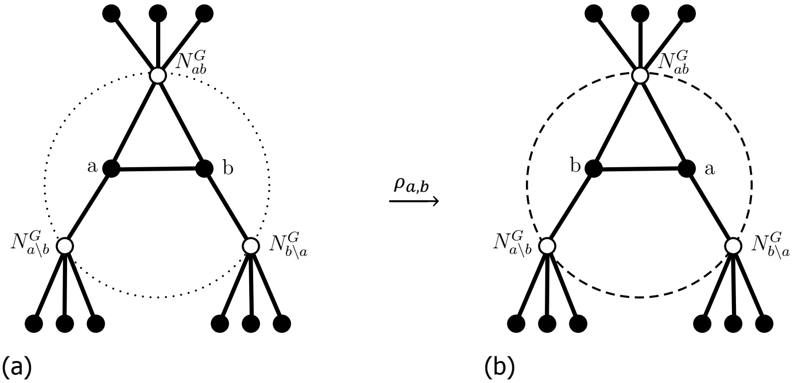


Figure 3.6: An example graph state before (left) and after (right) a pivot (ρ) on a and b. Note that a and b exchange neighbours. The dotted/dashed lines represent edges corresponding to two vertices from a different neighbourhood. This edges are complemented by the pivot operation. So for $x \in N^G_{ab}$, $y \in N^G_{a\backslash b}$ the edge (x, y) is complemented, i.e. deleted if it was present and added if it was not present.

Then we have that a pivot operation on (a, b) complements every edge (c, d) for which the vertices c, d are from a different neighbourhood of (a, b). Just as with local complementations, it is known which quantum gate(s) correspond(s) to a pivot. This is the topic of the following lemma.

Lemma 3.1. *Given a graph state* $|G\rangle$*. Assume that* $(a,b) \in E[G]$*. Then Hadamards on* a *and* b *corresponds to a pivot operation on* (a,b) *up to a Pauli-Z on* N^G_{ab}*, i.e.:*

$$Z[N^G_{ab}]H_a H_b\,|G\rangle = \left|\rho_{a,b}(G)\right\rangle \tag{3.15}$$

Proof. In this proof we will explicitly do the three local complementations corresponding to a pivot on a graph state and see how the corresponding quantum operations can be simplified. The total proof will consist of 4 steps, first three steps for a local complementation and the 4th to simplify the final expression. The same notation as in definition 3.3 is used for the sets of (common) neighbours.

- The first step is to see what operations are applied to $|G\rangle$ by the first local complementation. Therefore, we use that $N_a(G) = N^G_{ab} + N^G_{a\backslash b} + b$. Then from the definition of local complementation on graph states it follows that:

$$\left|\tau_a(G)\right\rangle = \sqrt{-iX}[a]\sqrt{iZ}[b]\sqrt{iZ}[N^G_{ab}]\sqrt{iZ}[N^G_{a\backslash b}]\,|G\rangle \tag{3.16}$$

- Using that $N^{\tau_a(G)}_b = a + N^G_{b\backslash a} + N^G_{a\backslash b}$, the second local complementation corresponds to the following operations:

$$\left|\tau_b\tau_a(G)\right\rangle = \sqrt{-iX}[b]\sqrt{iZ}[a]\sqrt{iZ}[N^G_{b\backslash a}]\sqrt{iZ}[N^G_{a\backslash b}]\,\left|\tau_a(G)\right\rangle \tag{3.17}$$

- For the last local complementation we use $N^{\tau_b\tau_a(G)}_a = b + N^G_{b\backslash a} + N^G_{ab}$ to find:

$$\left|\tau_a\tau_b\tau_a(G)\right\rangle = \sqrt{-iX}[a]\sqrt{iZ}[b]\sqrt{iZ}[N^G_{b\backslash a}]\sqrt{iZ}[N^G_{ab}]\,\left|\tau_b\tau_a(G)\right\rangle \tag{3.18}$$

- In the final step all these operations will be simplified to see what the effect is on $|G\rangle$.

$$\begin{aligned}\left|\tau_a\tau_b\tau_a(G)\right\rangle =&\sqrt{-iX}[a]\sqrt{iZ}[b]\sqrt{iZ}[N^G_{b\backslash a}]\sqrt{iZ}[N^G_{ab}]\,\left|\tau_b\tau_a(G)\right\rangle\\ =&(\sqrt{-iX}\sqrt{iZ})[a](\sqrt{iZ}\sqrt{-iX})[b](iZ)[N^G_{b\backslash a}]\sqrt{iZ}[N^G_{ab}]\sqrt{iZ}[N^G_{a\backslash b}]\,\left|\tau_a(G)\right\rangle\\ =&(\sqrt{-iX}\sqrt{iZ}\sqrt{-iX})[a](\sqrt{iZ}\sqrt{-iX}\sqrt{iZ})[b](iZ)[N^G_{b\backslash a}](iZ)[N^G_{ab}](iZ)[N^G_{a\backslash b}]\,|G\rangle\end{aligned} \tag{3.19}$$

Using $\sqrt{-iX}\sqrt{iZ}\sqrt{-iX} = \sqrt{iZ}\sqrt{-iX}\sqrt{iZ} = HY = iHXZ = -iHZX$ we find:

$$\left|\tau_a\tau_b\tau_a(G)\right\rangle = (-iHZX)[a](iHXZ)[b](iZ)[N^G_{b\backslash a}](iZ)[N^G_{ab}](iZ)[N^G_{a\backslash b}]\,|G\rangle \tag{3.20}$$

The next step is to identify some of these operations as stabilizers of $|G\rangle$. There are actually 2 of them. The first is $X_a Z_b Z_{N_{a\setminus b}} Z_{N_{ab}}$, the other is $X_b Z_a Z_{N^G_{b\setminus a}} Z_{N^G_{ab}}$. Using these both we find:

$$\begin{aligned} \left|\tau_a\tau_b\tau_a(G)\right\rangle &= (i)^3(-iH)[a](iH)[b](Z)[N^G_{ab}]\,|G\rangle \\ &= H_a H_b Z[N^G_{ab}]\,|G\rangle \end{aligned} \tag{3.21}$$

In the last linewe disregard global phase. As $|G\rangle$ is a graph state, $Z[N^G_{ab}]$ will flip the phase of every generator with a X as a position corresponding to a qubit in N^G_{ab}.

Thus, it follows that $\left|\rho_{a,b}(G)\right\rangle = H_a H_a\,|G\rangle$ when one disregards phases of the generators and global phase. □

There is one more quantum gate which we want to relate to graph operations. It is a little bit more trivial as it follows from the definition of graph states. The CZ gate on qubit i, j corresponds to flipping the edge (i,j): $CZ_{i,j}\,|G\rangle = \left|G + (i,j)\right\rangle$.

Next to gates on quantum states one can also do measurements. Here we will discuss Pauli-measurements. Lets start by introducing the following graph operations [18]:

$$Z_a(G) : G\setminus a \qquad \text{(remove } a \text{ from } G) \tag{3.22}$$

$$Y_a(G) : \tau_a(G)\setminus a \qquad \text{(local complement on } a \text{ and remove } a) \tag{3.23}$$

$$X_a(G) : \rho_a(G)\setminus a \qquad \text{(pivot on } a \text{ and remove } a) \tag{3.24}$$

This operations correspond to Pauli-X, Pauli-Y, Pauli-Z respectively measurements on vertex a up to single qubit Clifford operations. In figure 3.7 examples are given for how Pauli measurements on one qubit could act. In definition 3.4 the explicit expressions for measuring in a Pauli basis, including single qubit Clifford corrections are discussed. Therefore, the following notation for projectors is used:

$$P^a_{i,\pm} = |i,m\rangle\langle i,m|\,, \quad m = \begin{cases} 0 & \text{for } P^a_{i,+} \\ 1 & \text{for } P^a_{i,-} \end{cases} \tag{3.25}$$

where $i \in \{x,y,z\}$ and $m \in GF(2)$. For example, $P^a_{x,+} = |x,0\rangle\langle x,0| = |+\rangle\langle +|$. For convenience we write $Z[N]^m = \prod_{b\in N} Z^m_b$.

Definition 3.4. *The effect on the graph when measuring a qubit v of graph state $|G\rangle$ in Pauli-X,Y or Z basis is given as follows:*
Pauli-Z

$$|z,m\rangle\langle z,m|_v\,|G\rangle = \frac{1}{\sqrt{2}}\,|z,m\rangle_v\,Z[N_v(G)]^m\,|G\setminus v\rangle \tag{3.26}$$

$$= \frac{1}{\sqrt{2}}\,|z,m\rangle_v\,Z[N_v(G)]^m\,|Z_v(G)\rangle \tag{3.27}$$

Pauli-Y

$$|y,m\rangle\langle y,m|_v\,|G\rangle = \frac{1}{\sqrt{2}}\,|y,m\rangle_v\,\sqrt{(-1)^{1-m}iZ}[N_v(G)]\,|\tau_v(G)\setminus v\rangle \tag{3.28}$$

$$= \frac{1}{\sqrt{2}}\,|y,m\rangle_v\,\sqrt{(-1)^{1-m}iZ}[N_v(G)]\,|Y_v(G)\rangle \tag{3.29}$$

Pauli-X

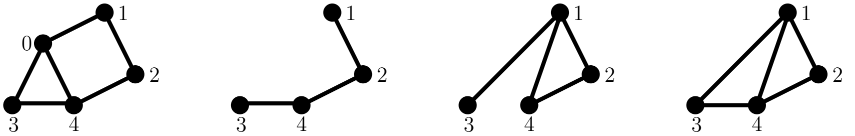
$$|x,m\rangle\langle x,m|_v\,|G\rangle = \frac{1}{\sqrt{2}}\,|x,m\rangle_v\,U^u_{x,m}\,|\tau_v\tau_u\tau_v(G)\setminus v\rangle \tag{3.30}$$

$$= \frac{1}{\sqrt{2}}\,|x,m\rangle_v\,U^u_{x,m}\,|X_a(G)\rangle \tag{3.31}$$

Where

$$U^u_{x,m} = \begin{cases} \sqrt{+iY}^u\, Z^{N_v\setminus(N_u\cup u)} & \textit{if } m = 0 \\ \sqrt{-iY}^u\, Z^{N_u\setminus(N_v\cup v)} & \textit{if } m = 1 \end{cases}$$

and $u \in N_v(G)$. If $|N_v(G)| = 0$, v is not a connected vertex so by definition in product state with the rest of the graph state. So the state on a is $|+\rangle$.



(a) The graph state $|G\rangle$ as shown in figure 3.1.

(b) Graph state up to single qubit Cliffords after measuring qubit 0 of $|G\rangle$ in the Z-basis.

(c) Graph state up to single qubit Cliffords after measuring qubit 0 of $|G\rangle$ in the Y-basis.

(d) Graph state up to single qubit Cliffords after measuring qubit 0 of $|G\rangle$ in the X-basis.

Figure 3.7: Overview of the effect of Pauli measurements on graph states up to single qubit Clifford operations.

It is proven by [10] that for a sequence of local complementation and vertex deletions, we can always do the deletion after the sequence of local complementation. This is useful when we for example have multiple Pauli-X measurements. In that case we can first do the sequence of local complementation and then remove the necessary vertices. This is formalized in the next lemma:

Lemma 3.2. *([10], Lemma 2.1) Let $G = (V, E)$ be a graph and $v, u \in V$ be vertices such that $v \neq u$, then*

$$\tau_v(G\backslash u) = \tau_v(G)\backslash u$$

3.3. Transforming graph states with single qubit Cliffords

In this section the single qubit Clifford equivalence of graph states is discussed. Before focussing on Clifford operations, we will start with a wider view. We distinguish single qubit unitary operations (SQU) and single qubit Clifford operations (SQC) . Two graphs, $G = (V, E)$ and $G' = (V, E')$, are called SQU-equivalent if there exists an $U \in \mathcal{U}_1^N$ such that this unitary maps $|G\rangle$ to $|G'\rangle$ ($|G'\rangle = U|G\rangle$). When we map the stabilizer $\mathcal{S}$ by this unitary the resulting set, $\{UsU^\dagger | s \in \mathcal{S}\}$, always stabilizes $|G'\rangle$, i.e. every element acts as identity on the state $|G'\rangle$. However, we are not sure that every element is in the Pauli-group. By definition (in section 2.2), a all elements of a stabilizer should be elements of the Pauli-group.

Let us consider the second type of operations we discussed, the single qubit Clifford group. We know that the Clifford group has the property that it maps the Pauli group to itself under conjugation, so they take stabilizers to stabilizers. When two graph states are equivalent under SQC, we will call this single qubit Clifford equivalent (SQC-equivalent). So $|G'\rangle = U|G\rangle$ for $U \in \mathcal{C}_1^N$. Next we will formally introduce the SQC-equivalence of quantum states.

SQC-equivalent

Definition 3.5. *Two quantum states $|\psi\rangle$, $|\psi'\rangle$ are single qubit Clifford (SQC) equivalent if and only if there exists an operation $U \in \mathcal{C}_1^N$ such that $U|\psi\rangle = |\psi'\rangle$.*

When only considering graph states, SQC-equivalence is defined as follows.

Definition 3.6. *Two graph states $|G\rangle$, $|G'\rangle$ are single qubit Clifford (SQC) equivalent if and only if there exists a symplectic matrix $\mathbf{Q}$ with the corresponding operation $U \in \mathcal{C}_1^N$ such that:*

$$(\mathbb{1}|\Gamma_G)\,\mathbf{P}\mathbf{Q}\,(\mathbb{1}|\Gamma_{G'}) = \mathbf{0} \tag{3.32}$$

When two graph states are SQC-equivalent we write this as:

$$|G\rangle \sim_{\text{SQC}} |G'\rangle \tag{3.33}$$

In literature, for example in [18], SQC-equivalent is called local Clifford (LC) equivalent. As this notation of locality conflicts with our notion of locality (defined in chapter 4), we will use SQC-equivalent instead. The same conflict holds for local unitary equivalent (LU) which is therefore called SQU-equivalent throughout this thesis. Locally equivalent graphs are called SQC-equivalent graphs in this research to avoid confusion in the usage of local. The reason why we discussed local complementations and their relation to SQC operations before, will now become very clear. Every SQC operation mapping graph states to graph states can be decomposed as a sequence of local complementations on the underlying graph [8]. The reverse is also true, as we have seen in definition 3.1 how a local complementation translates to single qubit Clifford operations. This leads to the following lemma, where LC refers to local complementations.

Theorem 3.1 ([8]).

$$G \sim_{LC} G' \iff |G\rangle \sim_{SQC} |G'\rangle \tag{3.34}$$

In the next section, we will formally introduce the problem of deciding SQC-equivalence of graph states and discuss the best known algorithm in terms of time complexity.

3.3.1. Definition and complexity

The decision problem whether two graph states are equivalent under SQC-operations we will call the single qubit Clifford equivalence (SQC-EQUIV) problem. SQC-EQUIV is formally introduced in problem 3.1.

Problem 3.1 (SQC-EQUIV). *Given graphs G, G' corresponding to graph states $|G\rangle$, $|G'\rangle$ on N vertices. Decide whether $|G\rangle$ and $|G'\rangle$ are SQC-equivalent.*

A polynomial time algorithm is found in [18]. This leads to theorem 3.2.

Theorem 3.2 ([18],). *The decision problem SQC-EQUIV is in $\mathbb{P}$. The running time of the algorithm is $\mathcal{O}\left(|G|^4\right)$.*

Proof from [18]. This is proven by Van Den Nest et al. [18], where a graph theory result from Bouchet is used [11]. The proof by Van Den Nest is repeated here.

From section 3.1.3 it is known that two stabilizer states $(\mathbf{X}, \mathbf{Z})$ and $(\mathbf{X}', \mathbf{Z}')$ are equivalent under Clifford operations if

$$(\mathbf{X}'|\mathbf{Z}')\mathbf{P}\mathbf{Q}(\mathbf{X}|\mathbf{Z})^T = \mathbf{0}\ . \tag{3.35}$$

For graph states these equations reduce to:

$$\Gamma'\mathbf{B}\Gamma + \mathbf{D}\Gamma + \Gamma'\mathbf{A} + \mathbf{C} = \mathbf{0} \tag{3.36}$$

For single qubit Clifford operations, $\mathbf{Q}$ can be written as four diagonal matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \tag{3.37}$$

Furthermore, for the diagonal matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$, the property that $\mathbf{Q}$ is a symplectic matrix ($\mathbf{Q}^T\mathbf{P}\mathbf{Q} = \mathbf{P}$) reduces to $\mathbf{AD} + \mathbf{BC} = \mathbb{1}$. Let $N = |G|$. By Gaussian elimination one finds a basis $\mathcal{B} = \{b_1, \dots, b_d\}$ of the solution space V for the linear equation 3.36. Note that equation 3.36 is a system of N^2 equations and $4N$ unknowns. For large N, this is usually a highly overdetermined system. Therefore, one expects the solution space of the linear equation to be relatively small. Unfortunately, it could be that $\dim \mathcal{B} = \mathcal{O}(N)$. This leads to an exponential number of vectors to check against the symplectic constraint (eq. 3.8). To avoid this, a result from graph theory by Bouchet is used [11]. This result states that, if $\dim \mathcal{B} > 4$, one only has to check all combinations of two basis vectors from $\mathcal{B}$ to find a solution if one exists. Therefore, the set of vectors V' that has to be considered if $\dim \mathcal{B} > 4$ is given as:

$$V' = \left\{ \sum_{i=1}^{\dim \mathcal{B}} x_i b_i : b_i \in \mathcal{B}, \quad x_i \in \{0,1\}, \quad \sum_{i=1}^{\dim \mathcal{B}} x_i \leq 2 \right\} \tag{3.38}$$

Note that $|V'| = 1 + \dim \mathcal{B} + \binom{\dim \mathcal{B}}{2}$. In the case that $\dim \mathcal{B} > 4$, the result can be stated as follows:

$$\exists \mathbf{A}', \mathbf{B}', \mathbf{C}', \mathbf{D}' \in V' : \mathbf{A}'\mathbf{D}' + \mathbf{B}'\mathbf{C}' = \mathbb{1} \Leftrightarrow \exists \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D} \in V : \mathbf{AD} + \mathbf{BC} = \mathbb{1} \tag{3.39}$$

Otherwise, if $\dim \mathcal{B} \leq 4$, all vectors in V have to be considered. This results in a running time of $\mathcal{O}(N^4)$ where $|G| = N$, as is shown in section 3.3.2. □

3.3.2. The algorithm

In this section we describe the algorithm proposed by van den Nest [8] in order to determine equivalence of graph states under SQC operations. The algorithm is given in pseudocode in algorithm 1 and an implementation in SAGE can be found on our GitHub [12]. The algorithm follows the steps of the proof of theorem 3.2 closely. In order to understand how the algorithm works, we will now discuss it line by line.

- line 1: We vectorize equation 3.36 using vec$(ABC) = (C^T \otimes A)$vec(B) which gives M, the vectorized form of $\mathbf{A}$,$\mathbf{B}$,$\mathbf{C}$,$\mathbf{D}$.
- line 2: AS described in the proof of theorem 3.2, $\mathbf{A}$,$\mathbf{B}$,$\mathbf{C}$,$\mathbf{D}$ are all diagonal matrices. Therefore, we only keep the rows of M corresponding to an element on the diagonal of $\mathbf{A}$,$\mathbf{B}$,$\mathbf{C}$,$\mathbf{D}$. This corresponds exactly to the rows $\{j + i +$

Algorithm 1 Solve SQC-EQUIV [8]

```
Input:   Graphs $G$ and $G'$ on the same vertex set $V$
Output:  TRUE if there exists a single qubit Clifford which transforms $|G\rangle$ to $|G'\rangle$
         FALSE otherwise
 1: $M \leftarrow [\mathbb{1}\otimes\Gamma', \Gamma\otimes\Gamma', \mathbb{1}\otimes\mathbb{1}, \Gamma\otimes\mathbb{1}]$      ▷$M$ is a $4N^2 \times N^2$ matrix.
 2: Keep only rows $\{jN^2 + i + iN\}_{i=0,j=0}^{N-1,3}$ of $M$, delete the other rows
 3:               ▷Keep only the diagonal elements of $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ in equation 3.37.
 4: $\mathcal{B} \leftarrow$ Solve$(M^T$, vec$(\mathbf{0}^{N\times N}))$
 5: if $\dim(\mathcal{B}) \leq 4$ then              ▷Bouchet's result on holds for $\dim(\mathcal{B}) \leq 4$
 6:     for $v \in span(\mathcal{B})$ do
 7:         if $v$ is a symplectic transformation then
 8:             return TRUE
 9: else
10:     for $v \in V'$ do                        ▷$V'$ is defined in eq. 3.38
11:         if $v$ is a symplectic transformation then
12:             return TRUE
13: return FALSE
```

$iN\}_{i=0,j=0}^{N-1,3}$. For example, if we consider the diagonal elements from $\mathbf{A}$, we find for $j = 0$ that we have $\{0, 1 + N, 2 + 2N, \dots, N^2 - 1\}$.

- line 3: The system of equations is solved to find a basis $\mathcal{B}$ of the solution space of equation 3.36.
- line 4-7: As Bouchet's result only holds for $\dim(\mathcal{B}) > 4$, we have to check all elements of the space spanned by $\mathcal{B}$. If an element if found which corresponds to a symplectic constraint, the algorithm returns TRUE.
- line 9-11: The algorithm only considers all elements in V', for which it checks if the corresponding transformation is symplectic. If this is the case, return TRUE.
- line 12: If none of the above returns TRUE, return FALSE.

We now continue by discussing the runtime of the algorithm.

- line 1: For every element of the new $4N^2 \times N^2$ matrix a multiplication is done (where multiplication is assumed to be constant time), so this line $\mathcal{O}(N^4)$.
- line 2: Here $4N^2 - 4N$ rows are removed, where every row is of length N^2. Thus, this line is also $\mathcal{O}(N^4)$. The result is a $4N \times N^2$ matrix M.
- line 3: Solving this system of equations is done using Gaussian elimination which, for M being a $4N \times N^2$ matrix, is done in $\mathcal{O}(N^4)$.

- line 4-7: The if statement is done in constant time. The loop runs over the full span of $\mathcal{B}$, which are in the worst case 2^4 elements. For every element, the constraint is calculated which makes the total runtime of this part $\mathcal{O}(2^4N)$.
- line 9-11: The for loop runs, in the worst case, over $|V'| = \mathcal{O}(N^2)$ elements. For every element $v \in V'$, the constraint has to be checked which is done in $\mathcal{O}(N)$. Therefore, the total for loop is of complexity $\mathcal{O}(N^3)$.

To summarize the running time of algorithm 1 is $\mathcal{O}(N^4)+\mathcal{O}(N^4)+\mathcal{O}(N^4)+\mathcal{O}(2^4N)+\mathcal{O}(N^3) = \mathcal{O}(N^4)$.

3.4. Reducing stabilizer states to graph states

It is shown [8] that every stabilizer state is single qubit Clifford equivalent to some graph state. This result is stated in theorem 3.3. In the proof of this theorem a constructive method is used to find the SQC operations which map a stabilizer state to a corresponding graph state. In this section we will discuss this method and apply it to an example. First we formally state the theorem.

Theorem 3.3 ([8]). *For every stabilizer state there exists a graph state which is single qubit Clifford equivalent to the stabilizer state and the running time of finding the corresponding graph state and the transformation taking the stabilizer state to the graph state is* $\mathcal{O}\left(|V[G]|^3\right)$.

To see how an equivalent graph state is found, we consider a general stabilizer state $S = (\mathbf{X}|\mathbf{Z})$. The goal is to find a symplectic matrix $\mathbf{Q}$ such that $(\mathbf{X}|\mathbf{Z})\mathbf{Q} = (\mathbf{X}'|\mathbf{Z}')$ where $\mathbf{X}'$ is an invertible matrix. Then it follows that ${\mathbf{X}'}^{-1}(\mathbf{X}'|\mathbf{Z}') = (\mathbb{1}|\Gamma)$ where Γ is an adjacency matrix with possible self connecting edges. These can be removed by an S operation on the qubits which have a self connecting edge. Then remaining phases can be corrected by a Z Pauli.

First we have to find $\mathbf{Q}$ such that $S\mathbf{Q} = S' = (\mathbf{X}'|\mathbf{Z}')$ has an X-part $\mathbf{X}'$ which is invertible. Note that $\mathbf{X}$ and $\mathbf{Z}$ are both $n \times n$ matrices and $\operatorname{rank}(S) = n$ by definition.

1. We start by doing Gaussian elimination to bring $(\mathbf{X}|\mathbf{Z})$ in the form:

$$\left(\begin{array}{c|c} \mathbf{R_x} & \mathbf{R_z} \\ \mathbf{0} & \mathbf{S_z} \end{array}\right)$$

   Where $\mathbf{R_x}$ has dimensions $k \times n$ and $k = \operatorname{rank}(\mathbf{R_x})$, $\mathbf{R_z}$ is also a $k \times n$ matrix and $\mathbf{S_z}$ is a $(n-k) \times n$ matrix. Note that Gaussian elimination has no physical effect on the state, it is only choosing a different set of generators.

2. Furthermore, the pivot columns of $\mathbf{R_x}$ are not always the first k columns. We now relabel the columns of $\mathbf{R_x}, \mathbf{R_z}, \mathbf{S_z}$ such that the pivot columns of $\mathbf{R_x}$ are listed as the first k columns. This leads to the following form:

$$\left(\begin{array}{cc|cc} \mathbf{R}^{\mathbf{1}}_{\mathbf{x}} & \mathbf{R}^{\mathbf{2}}_{\mathbf{x}} & {\mathbf{R_z}}' & \\ \multicolumn{2}{c|}{\mathbf{0}_{(n-k)\times n}} & {\mathbf{S_z}}^1 & {\mathbf{S_z}}^2 \end{array}\right)$$

Where $\mathrm{rank}({\mathbf{R_x}}^1) = k$ and $\mathrm{rank}({\mathbf{S_z}}^2) = n - k$.

3. If we now perform a Hadamard operation on qubits $k+1, \dots, n$, which maps $X \to Z$ and $Z \to X$ under conjugation, we find:

$$\left(\begin{array}{cc|cc} \mathbf{R}^{\mathbf{1}}_{\mathbf{x}} & \mathbf{R}^{\mathbf{2}}_{\mathbf{x}} & {\mathbf{R_z}}' & \\ \mathbf{0}_{(n-k)\times k} & {\mathbf{S_z}}^2 & {\mathbf{S_z}}^1 & \mathbf{0}_{(n-k)\times(n-k)} \end{array}\right) = (\mathbf{X}'|\mathbf{Z}')$$

The fact that $\mathbf{X}'$ is invertible follows from the fact that ${\mathbf{R_x}}^1$ and ${\mathbf{S_z}}^2$ are invertible. ${\mathbf{R_x}}^1$ is invertible by construction and the invertibility of ${\mathbf{S_z}}^2$ follows from $(\mathbb{1}|\Gamma)\mathbf{Q}(\mathbb{1}|\Gamma)^T = \mathbf{0}$ [8].

4. Again picking a different set of generators by multiplying the stabilizer state $(\mathbf{X}'|\mathbf{Z}')$ with ${\mathbf{X}'}^{-1}$ results in a stabilizer state $(\mathbb{1}|\Gamma)$ where Γ is an adjacency matrix with possible diagonal elements, corresponding to self connecting edges. The fact that Γ is an adjacency matrix (i.e. it is symmetric) follows from $(\mathbb{1}|\Gamma)\mathbf{Q}(\mathbb{1}|\Gamma)^T = \mathbf{0}$.

5. In step 2 a possible relabelling has happened. To undo this, we relabel the qubits/columns back to the original labelling.

6. To remove self connecting edges, we apply an S operation to every qubit with a self connecting edge. This removes the non zero diagonal elements of Γ, which makes Γ an adjacency matrix corresponding to a simple graph (graph state).

7. Finally, it could be that there are rows of the generator matrix with a -1 phase. Note that $\pm i$ is not possible, as the corresponding row would not be a stabilizer as $(\pm iP)^2 = -\mathbb{1}$ for $P \in \mathcal{P}^V$. The -1 phase can be cancelled by applying a Pauli-Z operation on the qubit with a X on the specific row. The -1 phase is cancelled, which follows from $ZXZ = -X$.

Note that the method of reaching a graph state only uses the following operations:

- Picking different sets of generators for the same state (in step 1 and 4)
- Hadamard (H) operations (step 3) and S operations (step 6)
- Relabelling of qubits (step 2 and 5), where the labelling before step 2 and after step 5 is the same.

- Z corrections for possible phases.

This method is implemented in Python in a toolbox for stabilizer states in SimulaQron [20]. In the remainder of this section the method described above will be used in an example.

Example: A graph state SQC-equivalent to GHZ_5

In this example we will find a graph state SQC-equivalent to the GHZ state on 5 qubits, which is given as:

$$|GHZ\rangle = \frac{|0\rangle^{\otimes 5} + |1\rangle^{\otimes 5}}{\sqrt{2}} \tag{3.40}$$

A generator matrix corresponding to this stabilizer state is:

$$\left(\begin{array}{ccccc|ccccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right) \tag{3.41}$$

We will now follow the method described above.

1. Gaussian elimination gives the following matrix:

$$\left(\begin{array}{ccccc|ccccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right) = \left(\begin{array}{c|c} \mathbf{R_x} & \mathbf{R_z} \\ \mathbf{0} & \mathbf{S_z} \end{array}\right) \tag{3.42}$$

   With $\mathbf{R_x} = [1,1,1,1,1]$, $\mathbf{R_z} = [0,0,0,0,0]$ and

$$\mathbf{S_z} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

2. The next step is to relabel the qubits such that the first k columns of $\mathbf{R_x}$ are the pivot columns. For this example, this is already the case so we continue.

3. In this step we perform a Hadamard operation on the last k qubits. The result is:

$$(\mathbf{X'}|\mathbf{Z'}) = \left(\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

4. As $\mathbf{X}'^{-1}$ exists, we can now multiply by this matrix to get an identity matrix as the X-part. Note that multiplying by $\mathbf{X}'^{-1}$ corresponds to a basis transformation. For intuition, one can also check that doing Gaussian elimination after the previous step also results in the following matrix.

$$\mathbf{X}'^{-1}\left(\mathbf{X}'|\mathbf{Z}'\right) = \left(\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{array}\right) = (\mathbb{1}|\Gamma)$$

5. Relabelling was not required in step 2, so relabelling is not needed here either.

6. The diagonal of Γ has all zero elements, so we can skip this.

7. All phases are +1, so no Pauli-Z operations are needed.

8. The result is the star graph $S_{\{0,1,2,3,4\},0}$ as shown in figure 3.8.

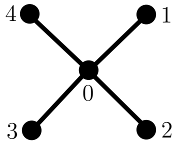


Figure 3.8: A star graph on 5 qubits.

This example shows that a GHZ-state is SQC equivalent to the star graph. By doing local complementation on the centre qubit, here qubit 0, we get the complete graph. By doing Hadamards on the leaves of the star graph, i.e. all qubits except the centre qubit, one obtains the GHZ state.

3.5. A quantum circuit in the graph state perspective

Here we discuss how the gate teleportation circuit proposed by Gottesman and Chuang [21] can be transformed to graph states and graph state operations. This will be used in chapter 5. The input of the original circuit, shown in figure 3.9, is a qubit a in state $|\psi\rangle$. The goal is to apply a gate $U \in \mathcal{U}(2)$ to the state $|\psi\rangle$ without doing the gate directly on qubit a. To achieve this goal, two ancilla qubits A_i for $i \in \{0, a\}$ are used. The result of the circuit is $U_a\,|\psi\rangle$, however the state is now on the ancilla qubit A_a instead of qubit a. The labelling is intentionally done such that the state of qubit a is transferred to the state of qubit A_a. Note that for $U_a = \mathbb{1}_2$ the circuit reduces to the well known teleportation circuit.

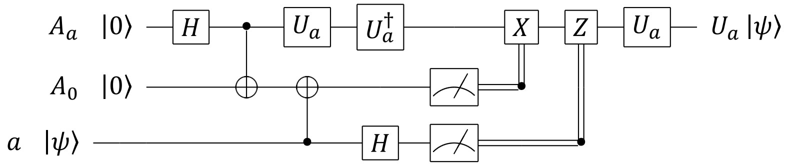


Figure 3.9: A circuit which achieves gate teleportation on a state $|\psi\rangle$ for gate U_a with two ancilla qubits A_a, A_0 prepared in a Bell state. [21]

As discussed in section 3.1 some Clifford operations correspond to known graph operations. It will be useful to rewrite the well known gate teleportation circuit in terms of known graph operations by using the following identity: $\mathrm{CNOT}_{i,j} = \mathrm{H}_i \mathrm{CZ}_{i,j} \mathrm{H}_i$. The circuit that follows is given below and will be very important in chapter 5.

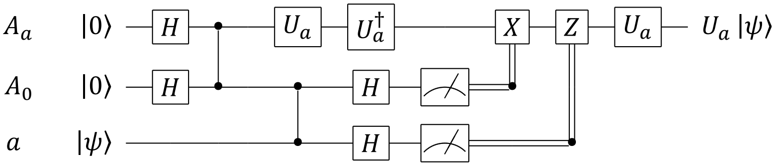


Figure 3.10: The gate teleportation circuit compiled with operations for which it is known how they transform graph states. Note that if $|\psi\rangle$ is a graph state and $U = \mathbb{1}$, this circuit can be fully described in the graph state picture.

In order to get a little bit more feeling for how the gate teleportation circuit works in the graph state picture, we will discuss an example of gate teleporting $U = \mathbb{1}_2$, i.e. regular qubit teleportation. In this example we have a graph state on 3 qubits corresponding to the complete graph and two ancilla qubits A_0,A_2. The circuit, shown in figure 3.11, teleports the state of qubit 2 to qubit A_2. In figure 3.12 the graph states at intermediate stages of the circuit are shown, where the stage corresponds to the labels in figure 3.11.

- (a) → (b): In this step two ancilla qubits, A_0 and A_2, are added to the graph state. In order to do this, the qubits are prepared in the $|+\rangle$ state. Note that the ancilla qubits are not yet entangled in the circuit, and not yet connected as a graph.
- (b) → (c): In this step a CZ gate is applied between A_0 and A_2, which is equal to adding an edge in the graph.
- (c) → (d): By a CZ between A_0 and qubit 2 of $|G\rangle$, the graph becomes connected.

- (d) → (e): From the previous step it is known that A_0 and 2 are connected, therefore using lemma 3.1 it follows that two Hadamards on adjacent vertices act as a pivot. From definition 3.2 the operations on the underlying graph are known: 1) A_0 and 2 exchange neighbours and 2) all edges corresponding to vertices from different neighbourhoods of $(2, A_0)$ are complemented. In this case, $N_{A_0\backslash 2} = A_2$, $N_{A_0 2} = \emptyset$ and $N_{2\backslash A_0} = 0, 1$. Therefore the edges $(0, A_2)$ and $(1, A_2)$ are complemented.

- (e)→(f): The qubits A_0 and 2 are measured in the Pauli-Z basis. Using the measurement rules from definition 3.4, we know that this corresponds to removing vertices from the graph.

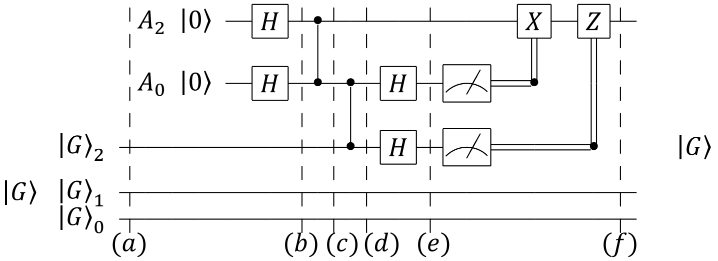


Figure 3.11: Teleporting the state of qubit 2 of $|G\rangle$ to qubit A_2. The dashed line labels refer to the subfigures in figure 3.12. So the dashed line labelled (a) corresponds to the graph state in figure 3.12(a). The input state and the output state are equal up to a vertex permutation.

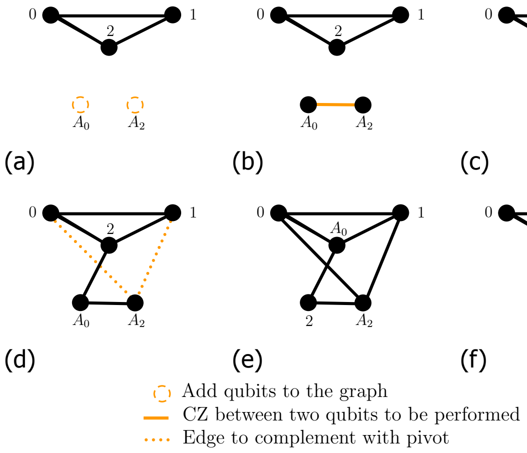

0
1
2
A_0
A_2
(a)
(b)
(c)
(d)
(e)
(f)
Add qubits to the graph
CZ between two qubits to be performed
Edge to complement with pivot

Figure 3.12: Step by step following the state of all 5 qubits when qubit 2 of a graph state $|G\rangle$ is teleported to another qubit A_2, as shown in figure 3.11. Orange is used to denote the effect of the next operation in the circuit. For example, the dashed circles in (a) represent the qubits added going from (a) to (b). The steps (d)->(e) and (e)->(f) are not completely one-to-one with the circuit, as in the graph state picture some single qubit Cliffords are disregarded.

4

Introducing local multi-qubit Clifford equivalence of graph states

This chapter takes the first steps in analysing local multi-qubit Clifford equivalence of graph states. To our knowledge, this topic has not been studied before, and therefore everything in this chapter is new. We start in section 4.1 with formally defining the problem. Furthermore, we discuss how this problem relates to known decision problems in some limiting cases (no multi-qubit nodes, all qubits in one node). We present an algorithm to decide T-LMQC equivalance which scales exponentially in the size of the graph in section 4.2. In this same section we also present a conjecture related to the presented algorithm. The final section of this chapter aims to give some feeling for which graph states are T-LMQC equivalent.

4.1. Introducing T-LMQC equivalence

In this section, first the notion of locality in graph states used in this thesis is defined (section 4.1.1). Afterwards, in section 4.1.2 the problem of deciding T-LMQC equivalence of graph states is formally introduced.

4.1.1. Locality in graph states

The topic of this section is to introduce the notion of locality with respect to graph states. By definition, every vertex in a graph state corresponds to a qubit. In a quantum internet, some of these qubits could be on the same quantum processor and other qubits could be on different quantum processors. When visualizing the quantum internet as a network, one also ends up at something like a graph - with nodes and with edges. However, every node in a quantum internet is a quantum processor, and a quantum processor could have access to multiple qubits. Therefore, we will make a distinction between nodes in a quantum internet and vertices/qubits in a graph state. To avoid confusion we will use the term node/-component for a quantum chip, and the term vertex or qubit for a node in a graph state. In other words, a node in a quantum internet is not equal to a node in a graph state. In order to describe the physical structure behind a graph state, we introduce a partition T of the vertex set of the graph state. T is defined as a tuple of components, such that every vertex in the graph state is in exactly one component. For example, for a graph state on 5 qubits where every (quantum internet) node has exactly 1 qubit, the partition is given as $T = (\{0\},\{1\},\{2\},\{3\},\{4\})$. However, when for example 0 and 1 share a node, the partition is given as $T = (\{0,1\},\{2\},\{3\},\{4\})$. We will use circles with dashed border and a grey filling in order to visualize the partition of a graph state, an example of this is shown in figure 4.1.

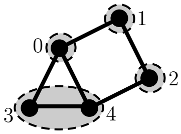


Figure 4.1: A graph $G = (V,E)$ where $V = \{0,1,2,3,4\}$ and $E = \{(0,1),(0,3),(0,4),(1,2),(2,4),(3,4)\}$ with partition $T = (\{0\},\{1\},\{2\},\{3,4\})$.

In this thesis we assume that every qubit in a node can be used to do two-qubit operations with at least one other qubit in the same node. Then we can use the fact with two-qubit operations a SWAP operation can be constructed, which can be used to swap around the qubits as needed. Then it follows that inside a quantum internet node, there is perfect connectivity of qubits. Note that, in order to achieve this experimentally, it is already sufficient if one can always make a path of two-qubit operations between two qubits inside a node. In figure 4.2 this is illustrated by an example.

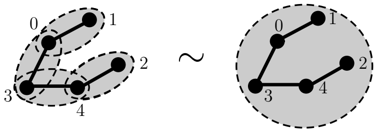


Figure 4.2: On the left hand side a graph state is shown where every two-qubit operations are possible between neighbours. As using two qubit operations, a SWAP operation can be constructed, the left hand side graph state is equivalent to the right hand side, where multi-qubit operations are clearly possible between any qubit in the node.

This leads to the following definition of local operations.

Local operations

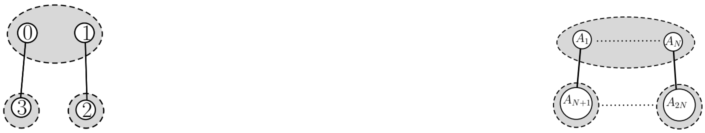
Definition 4.1. *Given a graph state $|G\rangle$ and a partition T of $V(G)$. An operation U is a local unitary operation w.r.t. to T if it can be written as a tensor product of operations acting inside the same node, i.e. $\bigotimes_{i\in T} U_i$ where $U_i \in \mathcal{U}(2^{|i|})$.*

In this thesis we focus on local Clifford operations, which we define as the operation C that can be written as $\bigotimes_{i\in T} C_i$ where $C_i \in \mathcal{C}^{|i|}$.

Local Clifford operations

Definition 4.2. *Given a graph state $|G\rangle$ and a partition T of $V(G)$. An operation U is a local Clifford operation w.r.t. T if it can be written as a tensor product of operations acting inside the same node, i.e. $\bigotimes_{i\in T} U_i$ where $U_i \in \mathcal{C}(2^{|i|})$.*

In the next section we will see how this translates to deciding equivalence of graph states under local Clifford operations. Below some more examples are given of graph states including visualization of locality.



(a) The graph state $|G_{\text{Bell }2}\rangle$ of two Bell pairs where both Bell pairs contribute one qubit to a node.

(b) The graph state $|G_{\text{Bell }N}\rangle$ of N Bell pairs where all Bell pairs contribute one qubit to a node.

4.1.2. Introducing T-LMQC equivalence

In this section the definition of local operations from the previous section is used in order to discuss equivalence of quantum states under local Clifford operations. This is formalized in the following definition:

T-LMQC-equivalence

Definition 4.3. *Given two quantum states $|\psi\rangle$, $|\psi'\rangle$ on N qubits, with a partition T of the qubits. $|\psi\rangle$ and $|\psi'\rangle$ are local multi-qubit Clifford equivalent w.r.t. T if and only if there exists a Clifford operation $U \in \mathcal{C}^N$ which is a local Clifford operation w.r.t. T such that:*

$$|\psi'\rangle = U\,|\psi\rangle \tag{4.1}$$

We will restrict ourselves to graph states in the following definition.

T-LMQC-equivalence

Definition 4.4. *Given two graph states $|G\rangle$, $|G'\rangle$ where $V[G] = V[G']$ with a partition T of the vertex set of G. Let Γ_G ($\Gamma_{G'}$) denote the adjacency matrix of G (G'). $|G\rangle$ and $|G'\rangle$ are local multi-qubit Clifford equivalent w.r.t. T if and only if there exists a Clifford operation $U \in \mathcal{C}^N$ which is a local Clifford operation w.r.t. T with the corresponding symplectic matrix $\mathbf{Q}$ such that:*

$$(\mathbb{1}|\Gamma_{G'})\mathbf{P}\mathbf{Q}(\mathbb{1}|\Gamma_G)^T = \mathbf{0} \tag{4.2}$$

Note equation 4.2 is equal to equation 3.9, it is repeated here for convenience. If two graph states are LMQC equivalent with partition T we write this as:

$$|G\rangle \sim_{T\text{-LMQC}} |G'\rangle \tag{4.3}$$

The problem of deciding T-LMQC equivalence of graph states is formally defined in problem 4.1.

Problem 4.1 (T-LMQC-EQUIV)**.** *Given graphs G, G' corresponding to graph states $|G\rangle$, $|G'\rangle$ with T a partition of the vertex set of G and G'. Decide whether $|G\rangle$ and $|G'\rangle$ are T-LMQC equivalent.*

For the single qubit case, it is known that deciding SQC equivalence corresponds to deciding LC equivalence of the underlying graphs (theorem 3.1). It would be interesting to also translate the T-LMQC-EQUIV problem to the underlying graphs and graph operations. In the following theorem we discuss that this is not trivial for T-LMQC-EQUIV. First, we need to introduce local edges.

Definition 4.5 (Local edges)**.** *Let G be a graph with a partition T of $V[G]$. A local edge of G w.r.t. T is an edge (u,v) such that $\exists t \in T : u,v \in t$. The graph operation of flipping a local edge will be referred to as LEF (local edge flips).*

Which enables us to state the following theorem.

Theorem 4.1. *Let G ($|G\rangle$) and G' ($|G'\rangle$) be two graph (states) on the same vertex set V, and let T be a partition of V. If G is equivalent to G' under local complementations (LC) and local edge flips (LEF) w.r.t T, then $|G\rangle$ and $|G'\rangle$ are T-LMQC equivalent.*

$$G \sim_{LC+T-LES} G' \Rightarrow |G\rangle \sim_{T-\text{LMQC}} |G'\rangle \tag{4.4}$$

Proof. From section 3.3 it follows that LCs correspond to SQC operations. From the definition of graph states, it follows that an edge flip in a graph corresponds to applying a CZ-gate on the underlying graph state. Local edge flips correspond to CZ-gates inside a multi-qubit node. As a CZ is a Clifford operation, it follows that any sequence of LC+LEF on a graph can be realized as a local multi-qubit Clifford on a graph state. The other way around is an open question, which we will discuss directly after this proof. □

Whether two graphs are equivalent under LC+LEF, if the corresponding graph states are T-LMQC equivalent, is an open question. One might think that a local Clifford U taking $|G\rangle$ to $|G'\rangle$ can be realized as a sequence of LC+LEF. However, U might be a sequence of multiple local Clifford operations, i.e. $U = U_m \otimes \cdots \otimes U_1$. Then we know that $|G'\rangle = U_m \otimes U_1 |G\rangle$. However, $U_1 |G\rangle$ in general is not a graph state. In other words, U may take a graph state to another graph state, but the intermediate states are not necessarily graph states. Therefore, we can in general not decompose U in a sequence of graph operations. However, we have not found an example of this.

For some T, T-LMQC-EQUIV can actually be solved in linear time. When there are no multi-qubit nodes, the T-LMQC-EQUIV problem reduces to SQC-EQUIV problem, which can be solved in linear time (theorem 3.2). On the other hand, it could be that all qubits share the same node, which implies that every edge can be deleted/added by only doing local operations. Therefore, every two graphs on the same vertex set are always locally equivalent. Thus, if all qubits share one node, T-LMQC-EQUIV can be solved in constant time ($\mathcal{O}(1)$). The interesting cases of T-LMQC-EQUIV are indeed somewhere in between these two cases, with (multiple) multi-qubit nodes and maybe single qubit nodes. Note that the number of local Clifford operations increases drastically with an increasing number of qubits per node. This follows from the fact that the size of the Clifford group increase exponentially. Already for nodes with a few qubits the number of allowed Cliffords is enormous. In section 2.2 the size of the Clifford group is discussed.

4.1.3. Local Clifford equivalence classes

We introduce the notion of local equivalence classes here. This is inspired on the definition of LC orbits used by Danielsen [22].

Definition 4.6. *Given a graph state $|G\rangle$ with partition T of $V(G)$. The local equivalence class $\mathbf{L}_T(G)$ is the set of all graph states, including $|G\rangle$, which can be reached from $|G\rangle$ by doing local Clifford operations with respect to T. I.e.:*

$$\mathbf{L}_T(G) = \{|G'\rangle : |G'\rangle \sim_{T\text{-LMQC}} |G\rangle\} \tag{4.5}$$

In the case of SQC operations the partition of $V(G)$ with $|V(G)| = N$ is given by $(\{1_1, \dots, 1_N\})$. For this case we will use the subscript 1 instead of T, i.e.:

$$\mathbf{L}_1(G) = \mathbf{L}_{(\{1_1,\dots,1_N\})}(G) = \{|G'\rangle : |G'\rangle \sim_{\text{SQC}} |G\rangle\} \tag{4.6}$$

It follows from the definition that SQC operations can not transform $|G\rangle$ to $|G'\rangle$ if $|G'\rangle \notin \mathbf{L}_1(G)$. This is different for T-LMQC operations, as there can $\exists G' \in \mathbf{L}_T(G)$ such that $G' \notin \mathbf{L}_1(G)$.

Definition 4.7. *The set of representatives of $|G\rangle$ is a set of graph states $rep(\mathbf{L}_T(G))$ such that for every $|G'\rangle$ T-LMQC equivalent to $|G\rangle$ there exists one graph state in the set which is SQC-equivalent to $|G'\rangle$.*

So $rep(\mathbf{L}_T(G)) = \{|G_1\rangle, \dots, |G_n\rangle\}$ where $|G_i\rangle \in \mathbf{L}_T(G)$ and $|G_i\rangle \nsim_{\text{SQC}} |G_j\rangle$ (not SQC-equivalent) for $i \neq j$ and $i, j \in \{1, \dots, n\}$. In this way a partition of $\mathbf{L}_T(G)$ is found: $\mathbf{L}_T(G) = \{\mathbf{L}_1(G_1), \dots, \mathbf{L}_1(G_n)\}$ such that all $\mathbf{L}_1(G_i) \in \mathbf{L}_T(G)$ are disjoint and that $\bigcup_i \mathbf{L}_1(G_i) = \mathbf{L}_T(G)$. $|rep(\mathbf{L}_T(G))|$ is the number of graph states in $rep(\mathbf{L}_T(G))$.

Example: Local Clifford equivalence classe of two Bell pairs

Given that $G = (\{0,1,2,3\}, \{(0,3),(1,2)\})$ and $T = (\{0,1\},\{2\},\{3\})$. Then using algorithm 2 we find that there are 4 representative graph states for $|G\rangle$, i.e. $|rep(\mathbf{L}_T(G))| = 4$.

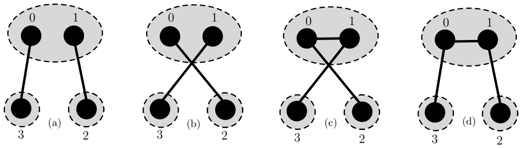


Figure 4.4: The four representatives of G in this example, where $G = G_1$ (there is no other graph state SQC equivalent to G). a) G_1, b) G_2, b) G_3, c) G_4.

With G_i for $i \in \{1,2,3,4\}$ given as in figure 4.4, the set of representatives is written as:

$$rep(\mathbf{L}_T(G)) = \{|G_1\rangle, |G_2\rangle, |G_3\rangle, |G_4\rangle\} \tag{4.7}$$

and the local equivalence class is given as:

$$\mathbf{L}_T(G) = \{\mathbf{L}_1(G_1), \mathbf{L}_1(G_2), \mathbf{L}_1(G_3), \mathbf{L}_1(G_4)\} \tag{4.8}$$

Note that $|G_i\rangle \in \mathbf{L}_T(G)$. Furthermore, two different representatives must be not-SQC equivalent: $|G_i\rangle \not\sim_{\text{SQC}} |G_j\rangle$ for $i \neq j$ and $i, j \in \{1, 2, 3, 4\}$. The operations needed to transform one representative of G to another are shown in figure 4.5.

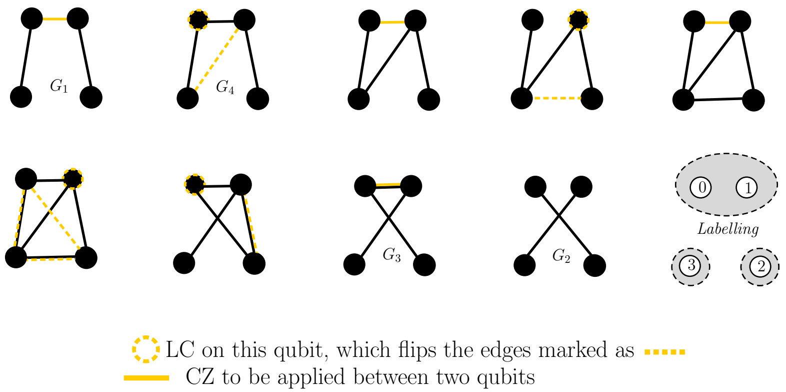

G_1
G_4
G_3
G_2
0
1
Labelling
3
2
LC on this qubit, which flips the edges marked as
CZ to be applied between two qubits

Figure 4.5: Going from any of the four representatives of $G = G_1$ to one of the others representatives. The orange markings indicate the action to take to go to the next graph state.

4.2. Brute force deciding T-LMQC equivalence

In this section we will describe an algorithm to solve T-LMQC-EQUIV. This algorithm will not be efficient, but it will provide tools to gain intuition for what graph states are T-LMQC equivalent. The first section relates local Clifford operations to local symplectic operations. The second subsection discusses the algorithm. In the last subsection a conjecture is presented which relates to the running time of the algorithm to solve T-LMQC-EQUIV.

4.2.1. Properties of local operations on graph states

In this section we will relate local Clifford operations to local symplectic operations. We start by discussing how to update a symplectic operation when applying extra

Clifford operations. I.e., for a given $U, U' \in \mathcal{C}^N$ with corresponding $\mathbf{Q}_U$, the question is how to find $\mathbf{Q}_{U'U}$ corresponding to $U'U$. For some Clifford operations it is known how it updates $\mathbf{Q}_U$. This is captured in the following definition, where we write sometimes write $\mathbf{Q}$ as a matrix of 4 matrices, i.e.:

$$\mathbf{Q} = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array}\right] = \begin{bmatrix} A_{11} & \cdots & A_{1N} & B_{11} & \cdots & B_{1N} \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ A_{N1} & \cdots & A_{NN} & B_{N1} & \cdots & B_{NN} \\ C_{11} & \cdots & C_{1N} & D_{11} & \cdots & D_{1N} \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ C_{N1} & \cdots & C_{NN} & D_{N1} & \cdots & D_{NN} \end{bmatrix}$$

Definition 4.8 ([23]). *Given $U \in \mathcal{C}^N$ with corresponding $\mathbf{Q}_U$. For $U' \in \{S_j, H_j, CZ_{i,j}\}$ with $i, j \in N$, the symplectic matrix $\mathbf{Q}_{U'U}$ follows from the following rules:*

- *$U' = S_j$: Column j of $\mathbf{Q}$ is added to column $N + j$.*

$$\begin{tikzcd} \cdots & A_{1j} \arrow[rr, bend left] & \cdots & B_{1j} & \cdots \\ \cdots & \vdots & \cdots & \vdots & \cdots \\ \cdots & C_{nj} & \cdots & D_{nj} & \cdots \end{tikzcd}$$

- *$U' = H_j$: Column j of $\mathbf{Q}$ is exchanged with column $N + j$.*

$$\begin{tikzcd} \cdots & A_{1j} \arrow[rr, bend left, leftrightarrow] & \cdots & B_{1j} & \cdots \\ \cdots & \vdots & \cdots & \vdots & \cdots \\ \cdots & C_{nj} & \cdots & D_{nj} & \cdots \end{tikzcd}$$

- *$U' = CZ_{i,j}$: Column j is added to column $N + i$ and column i is added to column $N + j$.*

$$\begin{tikzcd} \cdots & A_{1j} \arrow[rrrrrr, bend left] & \cdots & A_{1i} \arrow[rr, bend left] & \cdots & B_{1j} & \cdots & B_{1i} & \cdots \\ \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots & \vdots & \cdots \\ \cdots & C_{nj} & \cdots & C_{ni} & \cdots & D_{nj} & \cdots & D_{ni} & \cdots \end{tikzcd}$$

To give some intuition to how this works, we will now give an example of how Clifford operations relate to symplectic operations.

Example: Graph state initialization

In this example we transform the $|00\rangle$ into a connected graph state on 2 qubits and discuss the corresponding symplectic transformation. We start with two qubits in the $|0\rangle$ state, which is trivially stabilized be the Z operator so a generator matrix is given by:

$$(X_0|Z_0) = \left(\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right)$$

Following our definition of graph states, we first have to apply a H on both qubits and then do a CZ operation. We know how these operations update a symplectic transformation, but there is no symplectic transformation to update yet. Therefore, we start from the identity operation $\mathbb{1}_{2N}$. Then first the two Hadamards are applied. According to definition 4.8, for a Hadamard on qubit j this corresponds to exchanging column j and column $j+N$. Using this we find:

$$Q_{\text{idle}} = \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right) \xrightarrow{H\otimes H} Q_{H\otimes H} = \left(\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right)$$

When we would transform $(X_0|Z_0)$ by $\mathbf{Q}_{H\otimes H}$ the resulting generator matrix is a graph state without any edges:

$$(X_0|Z_0)\left(\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right) = \left(\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right)$$

The next step is to do a CZ between the two qubits, for which we find the corresponding update to $\mathbf{Q}_{H\otimes H}$ using definition 4.8. The result is:

$$\mathbf{Q}_{H\otimes H} = \left(\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array}\right) \xrightarrow{CZ} Q_{CZ(H\otimes H)} = \left(\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{array}\right)$$

Applying $Q_{CZ(H\otimes H)}$ to $(X_0|Z_0)$ results in the generator matrix of the complete graph on 2 qubits.

$$(X_0|Z_0)\left(\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{array}\right) = \left(\begin{array}{cc|cc} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{array}\right)$$

This indeed corresponds to a graph state related to the complete graph on 2 vertices.

Definition 4.8 will be very important for the following lemma.

Lemma 4.1. *Given a graph state $|G\rangle$ with related graph $G = (V,E)$ and a partition*

T of this vertex set V. Assume that the vertices in T are ordered such that for every part, the vertices are consequitive. I.e., a part can be $\{6,7,8\}$ but it can not be $\{0,5,10\}$. For any local Clifford operation C there exists a corresponding symplectic matrix $\mathbf{Q} = [[\mathbf{A},\mathbf{B}],[\mathbf{C},\mathbf{D}]]$ where

$$\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D} \in \{\bigoplus_{i\in T}\mathbf{M}_i \ : \ \mathbf{M}_i \in \mathbb{F}_2^{|i|\times|i|}\} \tag{4.9}$$

such that $\mathbf{QPQ}^T = \mathbf{P}$. The $\bigoplus$ sign is the direct sum as defined in section 2.1.

Note that local Clifford operations are found using the tensor product ($\otimes$) and the symplectic transformations are found using the direct sum ($\bigoplus$), i.e.:

$$C = \bigotimes C_i \quad \sim \quad \bigoplus \mathbf{M}_i = \mathbf{M}$$

This is not a formal relation, but it might help to give some intuition to how the symplectic operations are constructed.

Proof. The proof of lemma 4.1 consists of the following steps. During this proof we will keep track of the possible non-zero elements of the matrices $\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}$, in order to show that these matrices indeed satisfy equation 4.9. The possible non-zero positions of these four matrices are given by the matrices NZ_A, NZ_B, NZ_C, NZ_D, which are all $N \times N$ matrices with elements in $\mathbb{F}_2$. We start from the possible non-zero positions of the idle circuit, and then continue to update them according to SQC and T-LMQC operations. At some point, adding more operations will not change the positions any further.

1. The idle circuit is given by four $N \times N$ matrices $\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}$, where $\mathbf{A},\mathbf{D} = \mathbb{1}_N$ and $\mathbf{B},\mathbf{C} = \mathbf{0}$. Therefore, the possible non-zero elements are given by:

   $$NZ_A = \mathbb{1}_N, \quad NZ_B = \mathbf{0}, \quad NZ_C = \mathbf{0}, \quad NZ_D = \mathbb{1}_N$$

   By doing SQC operations, any combination of S and H gates, we use the rules from definition 4.8 to find that the possible non-zero positions are given by:

   $$NZ_A = \mathbb{1}_N, \quad NZ_B = \mathbb{1}_N, \quad NZ_C = \mathbb{1}_N, \quad NZ_D = \mathbb{1}_N$$

2. In this step the possible non-zero positions are updated after doing T-LMQC operations. Therefore, we assume that the columns of $\mathbf{Q}$ are labelled such that qubits in the same node are adjacent. If this is not the case, this can be achieved by relabelling columns (up to row swaps), which are not physical operations. Using the rule for a CZ gate, we find that a CZ gate makes the NZ matrices block diagonal, where the dimensions of every block correspond to the number of qubits in the node. Here $\mathbf{J}_N$ is used for the $N \times N$ all-ones matrix.

   $$NZ_A = \bigoplus_{i\in T}\mathbf{J}_{|i|}, \quad NZ_B = \bigoplus_{i\in T}\mathbf{J}_{|i|}, \quad NZ_C = \bigoplus_{i\in T}\mathbf{J}_{|i|}, \quad NZ_D = \bigoplus_{i\in T}\mathbf{J}_{|i|}$$

3. Any extra S, H or CZ operation will not change the possible non-zero positions any more. This follows again from the rules in definition 4.8. This proves that any symplectic transformation corresponding to a sequence of H, S and CZ gates, will satisfy equation 4.9. Furthermore, as the set $\{H, S, CZ\}$ generates the Clifford group, any local Clifford corresponds to a symplectic matrix described by equation 4.9. This concludes the proof.

□

An example for the form of $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ corresponding to $\mathbf{Q}$ with partition $T = (\{0,1\}, \{2,3,4\}, \{5,6\})$ is given below where $m_i \in \{0,1\}$ for $i \in \{0, \ldots, 16\}$:

$$\begin{bmatrix} \begin{bmatrix} m_0 & m_1 \\ m_2 & m_3 \end{bmatrix} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \begin{bmatrix} m_4 & m_5 & m_6 \\ m_7 & m_8 & m_9 \\ m_{10} & m_{11} & m_{12} \end{bmatrix} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \begin{bmatrix} m_{13} & m_{14} \\ m_{15} & m_{16} \end{bmatrix} \end{bmatrix} \tag{4.10}$$

4.2.2. Overview of the algorithm

In order to gain more intuition about which graph states are equivalent under local multi-qubit Clifford operations we used an algorithm which is not efficient. This algorithm is described in Algorithm 2 and can be found on Github [12]. In this subsection we will discuss this algorithm step-by-step and its correctness. We first state the algorithm:

The goal of the algorithm is to determine whether two graph states are equivalent under local multi-qubit operations with respect to T. The input to the algorithm is two graphs, G and G', and a partition T of the vertex set of the graphs G and G'. The algorithm is restricted to two graphs on the same vertex set and the same partition. We will discuss the algorithm line by line.

- line 1: From vectorizing linear equation 3.36 we find M.
- line 2: This is the step where local operations are enforced. By only selecting the rows which correspond to possible non-zero values in $\bigoplus_{i \in T} \mathbf{J}_{|i|}$, the other elements of $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ must be zero. This follows from lemma 4.1.
- line 4: The remaining system of equations is solved to find a basis $\mathcal{B}$.
- line 5-8: Every element in the space spanned by $\mathcal{B}$ is checked if it is symplectic. If it is symplectic, the algorithm returns TRUE

Algorithm 2 T-LMQC-EQUIV (brute-force)

```
Input:  Graphs $G$ and $G'$ on the same vertex set $V$ and a partition $T$ of $V$.
        $T$ must be ordered such that qubits in the same node are also
        consecutive (up to other qubits in the same node)
Output: TRUE if there exists a local Clifford which transforms $G$ to $G'$
        FALSE otherwise
1: $M \leftarrow [\mathbb{1} \otimes \Gamma', \Gamma \otimes \Gamma', \mathbb{1} \otimes \mathbb{1}, \Gamma \otimes \mathbb{1}]$
2: Only keep rows of $M$ corresponding to a 1 in vec($\bigoplus_{i \in T} \mathbf{J}_{|i|}$)
3:         ▷The remaining matrix is of dimensions $4m \times N^2$ where $m = \sum_{i \in T} |i|^2$.
4: $\mathcal{B} \leftarrow$ Solve($M^T, \mathbf{0}^{4m \times 1}$)
5: for $v \in span(\mathcal{B})$ do
6:     Find $\mathbf{E}, \mathbf{F}, \mathbf{G}, \mathbf{H}$ corresponding to $v$
7:     if $[[\mathbf{E}, \mathbf{F}], [\mathbf{G}, \mathbf{H}]]$ is symplectic then
8:         return TRUE
9: return FALSE
```

- line 9: If no symplectic operation satisfying equation 3.36 is found, the algorithm returns FALSE.

The next thing to do is analyse the time complexity of this algorithm. Therefore, we will analyse the algorithm line-by-line.

1. Line 1-2: This is constructing a $4N^2 \times N^2$ matrix and then removing some rows, which is done in $\mathcal{O}(N^4)$.

2. Line 3: Solving this system of equations can be done using Gaussian elimination. The dimensions of M are $4m \times N^2$, and as $m = \mathcal{O}(N^2)$, we find that this step can be done in $\mathcal{O}(m^2 N^2) = \mathcal{O}(N^6)$.

3. Line 4-7: This loop runs over 2^m elements, where $m = \sum_{i \in T} |i|^2$. For every iteration of the for loop, matrix multiplications is done of matrices of dimension $N \times N$, which is done in $\mathcal{O}(N^3)$. So the full body has time complexity $\mathcal{O}(2^m N^3)$.

Thus, the time complexity of algorithm 2 is $\mathcal{O}(N^4) + \mathcal{O}(N^6) + \mathcal{O}(2^m N^3) = \mathcal{O}(2^m N^3)$. This leads us to the theorem corresponding to this algorithm.

Theorem 4.2. *Given $|G\rangle$, $|G'\rangle$ and a partition T of the vertex set of $V[G]$ $(= V[G'])$. Algorithm 2 returns true if and only if $|G\rangle \sim_{T-\text{LMQC}} |G'\rangle$ and returns false otherwise. The running time of this algorithm is $\mathcal{O}(2^m N^3)$ where $m = \sum_{i \in T} |i|^2 = \mathcal{O}(N^2)$.*

4.2.3. Conjecture

In this section we present a conjecture which, if true, improves the run time of algorithm 2. To be able to state the conjecture, we first formally discuss locally connected graphs.

Definition 4.9 (Locally connected graphs)**.** *Let G be a graph with a partition T of $V[G]$. Furthermore, let G' be the graph which results from adding every local edge to G which is not in G. Then a graph G is a locally connected graph w.r.t. T if and only if G' is a connected graph.*

The conjecture is very similar to the result of Bouchet which is used to find a polynomial time algorithm for SQC-EQUIV [11]. The restriction on locally connected graphs is necessary, as counter-examples are otherwise found when testing the conjecture.

Conjecture

Conjecture 4.1. *Given $|G\rangle$, $|G'\rangle$ and a partition T of the vertex set of $V[G]$ ($= V[G']$). Let the maximum number of qubits per node be upper bounded by 2. Furthermore, we assume G and G' are locally connected graphs. Lastly, we assume that no local edges are present in G and G'. Let $\mathcal{B}$ be a basis of the solution space of equation 3.36. Furthermore, t_2 is the number of two-qubit nodes in T. In order to verify that a solution in span($\mathcal{B}$) is a symplectic operation, it is sufficient to check if there exists a solution in V' which is symplectic. In other words:*

$$\begin{gathered}\exists v(\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}) \in V : \mathbf{AD}+\mathbf{BC}=\mathbb{1} \\ \Leftrightarrow \\ \exists v(\mathbf{A}',\mathbf{B}',\mathbf{C}',\mathbf{D}') \in V' : \mathbf{A}'\mathbf{D}'+\mathbf{B}'\mathbf{C}'=\mathbb{1}\end{gathered} \tag{4.11}$$

where

$$V' = \left\{\sum_{i=1}^{|\mathcal{B}|} x_i b_i : b_i \in \mathcal{B}, x_i \in \{0,1\}, \sum_{i=1}^{|\mathcal{B}|} x_i \leq 2+2t_2\right\} \tag{4.12}$$

The size of V' is given as follows:

$$|V'| = \sum_{i=1}^{2+2t_2} \binom{M}{i}, \qquad M = \sum_{i\in T} |i|^2 \tag{4.13}$$

Note that if $t_2 = 0$, the conjecture reduces to the result from Bouchet [11] used in theorem 3.2. Using $M = \sum_{i\in T} |i|^2 = 2^2 t_2 + |V| - 2t_2 = 2t_2 + |V|$, an upper bound on the size of V' can be found:

$$|V'| = \sum_{i=1}^{2+2t_2} \binom{M}{i} \leq \sum_{i=1}^{2+2t_2} M^i = \sum_{i=1}^{2+2t_2} (2t_2 + |V|)^i \tag{4.14}$$

Therefore, we know that $|V'|$ scales polynomial in $\left|V[G]\right|$ for a fixed t_2:

$$\mathcal{O}(|V'|) = \mathcal{O}\left((2t_2 + |V|)^{2+2t_2}\right) \tag{4.15}$$

Conjecture 4.1 has been tested for all possible combinations of G and G' for $|G| \leq 5$ and for $t_2 = 1, 2$, and also for $|G| = 6$ and for $t_2 = 1$. This is illustrated by the possible vertex sets and partitions in figure 4.6. If the conjecture is true, this can be used to find an algorithm with an improved runtime to solve T-LMQC-EQUIV, this is the topic of the next theorem.

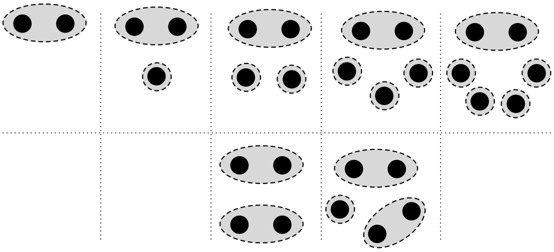


Figure 4.6: Conjecture 4.1 is tested for all possible locally connected graph states $|G\rangle$, $\left|G'\right\rangle$ on vertex set V and partition T shown here.

Theorem 4.3 (if conjecture 4.1 is true)**.** *Given $|G\rangle$, $|G'\rangle$ and a partition T of the vertex set of $V[G]$ ($= V[G']$). Assume the qubits per node is bounded by 2 and the number of two-qubit nodes is given by t_2. Furthermore, assume that G and G' are locally connected graphs. Algorithm 3 returns true if and only if $|G\rangle \sim_{T\text{-LMQC}} |G'\rangle$ and returns false otherwise. The running time of this algorithm is $\mathcal{O}\left(\sum_{i=1}^{2+2t_2} \binom{m}{i} N^3\right)$ where $m = \sum_{i \in T} |i|^2$.*

Proof. The algorithm is identical to the proof of algorithm 2, except line 7. Instead of checking all vectors in V, the algorithm only checks all vectors in V'. Conjecture 4.1 states that checking V' is sufficient to find a solution, if such a solution exists in V. □

Here we state the algorithm based on conjecture 4.1. Note that the algorithm is very similar to algorithm 2. We will discuss the lines different from algorithm 2.

Algorithm 3 Check multi-qubit Clifford equivalence (conjecture-brute-force)

```
Input:  Graphs $G$ and $G'$ on the same vertex set $V$ and a partition $T$ of $V$,
        such that $\max_{i \in T}(i) \leq 2$. Furthermore, $G$ and $G'$ are
        locally connected graphs w.r.t. $T$.
Output: TRUE if there exists a local Clifford $\mathbf{Q}$ which transforms $|G\rangle$ to $|G'\rangle$
        FALSE otherwise
 1: remove all local edges (w.r.t. $T$) in $G$ and $G'$
 2: $M \leftarrow [\mathbb{1} \otimes \Gamma', \Gamma \otimes \Gamma', \mathbb{1} \otimes \mathbb{1}, \Gamma \otimes \mathbb{1}]$
 3: Only keep rows of $M$ corresponding to a 1 in vec($\bigoplus_{i \in T} \mathbf{J}_{|i|}$)
 4:         ▷The remaining matrix is of dimensions $4m \times N^2$ where $m = \sum_{i \in T} |i|^2$.
 5: $\mathcal{B} \leftarrow$ Solve($M^T, \mathbf{0}^{4m \times 1}$)
 6: for $v \in V'$ do                                ▷$V'$ is defined in conjecture 4.1
 7:     Find $\mathbf{E}, \mathbf{F}, \mathbf{G}, \mathbf{H}$ corresponding to $v$
 8:     if $[[\mathbf{E}, \mathbf{F}], [\mathbf{G}, \mathbf{H}]]$ is symplectic then
 9:         return TRUE
10: return FALSE
```

- line 1: All local edges are removed, because this is one of the requirements of the conjecture.

- line 6: Instead of $span(\mathcal{B})$ the algorithm only considers all elements in $V' \subseteq span(\mathcal{B})$, where V' is defined in conjecture 4.1.

Algorithm 3 is implemented in SAGE, the code can be found on Github [12] and examples using this code can be found in appendix A.1. In figure 4.7 the running time of algorithm 2 and algorithm 3 are shown.

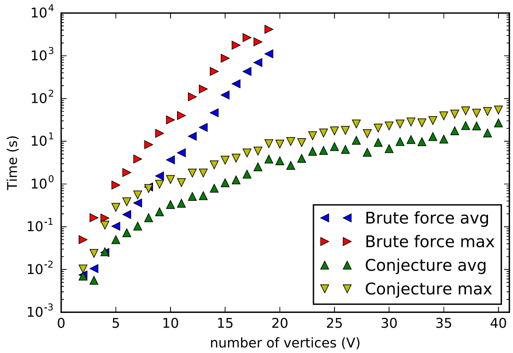


Figure 4.7: Maximum and average running times of algorithm 2 ("Brute Force") and algorithm 3 ("Conjecture"). The input graphs are randomly generated connected graphs and the number of two-qubit nodes is one.

4.3. Examples of T-LMQC equivalent graphs

Algorithm 2 solves T-LMQC-EQUIV, as this is a new result we will here present some examples of T-LMQC equivalent graph states.This hopefully helps to get some intuition for local Clifford equivalence. The first example is to check whether the GHZ state (complete graph) on 4 vertices and two Bell pairs sharing a node are T-LMQC equivalent. It would be very useful if this was true, as Bell pairs are often created in experimental setups and the GHZ-state is often used as an input for network protocols. Unfortunately, it turns out that the two are not equivalent under local multi-qubit operations:

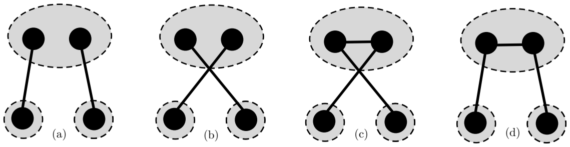


Figure 4.8: All graphs which are equivalent up to SQC to $|G_{\text{Bell2}}\rangle$ under LMQC operations given $T = (\{0,1\},\{2\},\{3\})$, i.e. $\mathbf{L}_T(G_{\text{Bell }2})$. As $\left|rep(\mathbf{L}_T(G_{\text{Bell }2}))\right| = 4$, we have 4 graph states here.

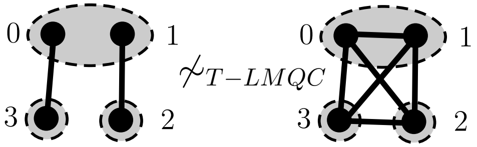


Figure 4.9: Using algorithm 2 we find that two Bell pairs are not T-LMQC equivalent to the complete graph, with partitions as in this figure.

The next example considers all graph states on 4 vertices. We will check which of them are equivalent under local multi-qubit operations if there is one two-qubit node. It turns out that the 18 single qubit Clifford equivalence classes form 6 T-LMQC equivalence classes (given $T = (\{0,1\},\{2\},\{3\})$) as shown in figure 4.10.

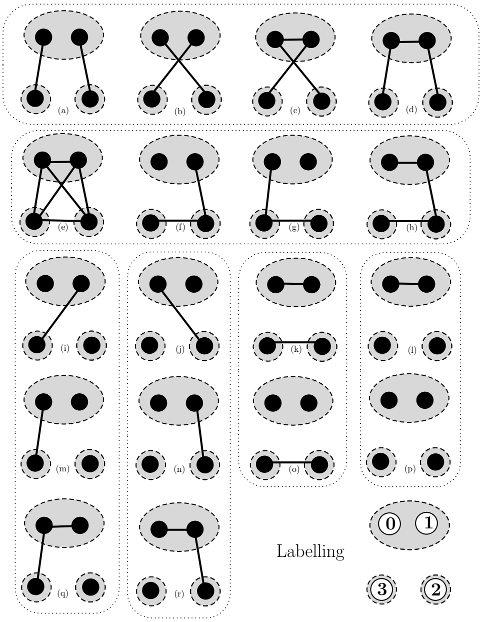


Figure 4.10: Representatives of all single qubit Clifford equivalence classes on four qubits and with $T = (\{0,1\},\{2\},\{3\})$. Note that only (a)-(h) are (can be) connected graphs on four qubits. The dotted lines indicate different LMQC equivalence classes. For example, the four graphs (a)-(d) are not SQC-equivalent but they are LMQC equivalent. And the graphs in figure (a) and (e) are not SQC-equivalent and not LMQC-equivalent.

5

Mapping multi-qubit operations to ancilla states

In this chapter we propose a new algorithm to decide local multi-qubit Clifford equivalence of graph states. The concept behind this algorithm is not restricted by the number of qubits per node, however we will focus on the algorithm for maximum 2 qubits per node. The algorithm still scales exponentially in the number of multi-qubit nodes, but linearly in the size of the graph state. As the concept behind this algorithm is completely new, we will try to give some intuition for why it works in section 5.1. In section 5.2 we formalize the concept of the first section and provide the main result of this chapter, theorem 5.1. This theorem is proved in section 5.3. Section 5.4 provides lemmas for the proof of theorem 5.1. However, some results in section 5.4 might be of interest outside the context of theorem 5.1. In section 5.5 we finally present the algorithm, which is relatively straightforward when the result of section 5.2 is understood. In section 5.5.4 we compare the algorithm proposed in this chapter with the brute force algorithm proposed in chapter 4. Finally, in section 5.6 we discuss how the algorithm would work for general multi-qubit nodes.

5.1. Introducing deciding T-LMQC equivalence via gate teleportation

The naive approach of checking local multi-qubit Clifford equivalence of graph states is to apply every possible local Clifford to the source state and then verify whether the target state is reached. However, even for small multi-qubit nodes the size of the local Clifford group is enormous. For example, as discussed in chapter 2, the two qubit Clifford group has 11520 elements. Nevertheless, for deciding single qubit Clifford equivalence of graph states a polynomial time algorithm is known (see 3.1). To optimize the naive approach a little bit, one could first consider all local multi-qubit operations and then use the polynomial time algorithm instead of checking all single qubit Cliffords. To be more specific, given two graph states $|G\rangle$ and $|G'\rangle$ on N qubits for which one wants to decide T-LMQC equivalence. Lets assume, for example, that there are two two-qubit nodes, A and B, and all other nodes are single qubit nodes. Then we apply every local multi-qubit operation to the source state $|G\rangle$ to find the set $G_{all} = \{C_A \otimes C_B\, |G\rangle \,|C_A, C_B \in \mathcal{C}_2\}$. This is a set of possibly 11520×11520 stabilizer states, as C_A and C_B can both be any element from $\mathcal{C}_2$ and $|\mathcal{C}_2| = 11520$. Then, for every stabilizer state in G_{all} we check SQC-equivalence with $|G'\rangle$. This can either be done by considering every SQC operation per node or by using the efficient algorithm. The naive approach would lead to considering $24^{N-|A|-|B|}$ single qubit Clifford operations, where the optimized naive approach would scale linearly with N: $\mathcal{O}(N^4)$.

In this chapter we provide an algorithm to decide T-LMQC equivalence which is very similar to the improved naive approach. The difference is that, instead of considering every element of the local multi-qubit Clifford group, considering a smaller set (the reduced Clifford set, $\mathcal{C}_2'$) is sufficient. The reduced Clifford set is formally defined in section 5.2. A rough sketch of the algorithm is as follows. The algorithm will first generate a set of stabilizer states, $G'_{all} = \{C_A, C_B \in \mathcal{C}_2' | C_A \otimes C_B\, |G\rangle\}$, and then check SQC-equivalence to $|G'\rangle$ for every graph state in G'_{all}. The improved runtime is due to the reduced size of $\mathcal{C}_2'$: $|\mathcal{C}_2'| \ll |\mathcal{C}_2|$.

The big question remaining is how to find this reduced Clifford set $\mathcal{C}_2'$. The approach discussed in this chapter is based on the gate teleportation circuit discussed in section 3.5. Here we will try to introduce the concept informally, which hopefully provides some intuition for this approach. In the next section a more formal discussion is provided.

Lets start by studying the gate teleportation circuit in a bit more detail. In order to teleport a M-qubit gate $U_M \in \mathcal{C}(2^M)$, $2M$ ancilla qubits are needed. We will refer to the quantum state of these qubits as the ancilla state. The ancilla qubits must be initialized in a specific quantum state $|A\rangle$. The state $|A\rangle$ is known, it is always the M-Bell pairs graph state. An example for $M = 5$ is shown in figure 5.1. The next step is to apply U_M to the ancilla state instead of applying it directly to $|G\rangle$. An example

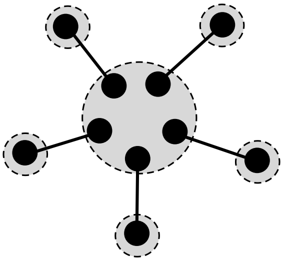
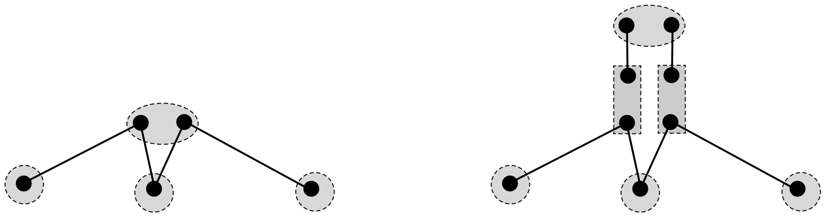


Figure 5.1: The M-Bell pairs state for $M = 5$, where every Bell pair contributes one qubit to a multi-qubit node.

of this is shown in figure 5.2, for the case that $|G\rangle$ is a graph state on 5 vertices with 1 two qubit node. The state $U_M |A\rangle$ might not be a graph state, as not every Clifford maps graph states to graph states, but it is definitely a stabilizer state as U_M is a Clifford. As discussed in section 3.4 a stabilizer state can always be transformed to a graph state with only SQC-operations. In other words, starting from the M-Bell pairs graph state, MQC operations are applied on M qubits and SQC-operations on $2M$ qubits such that the resulting state is a graph state. In terms of local operations, this corresponds to local operations with partition $T_A = (\{0, \dots, M-1\}, \{M\}, \dots, \{2M-1\})$. U_M is applied to the node with qubits $\{0, \dots, M-1\}$, after which SQC-operations on possibly all ancilla qubits are applied to reach a graph state. Using algorithm 2 we can find all graph states T_A-LMQC to $|A\rangle$, i.e. $\mathbf{L}_{T_A}(A)$. At first sight it might not be clear why would want to do this. Let us consider the case for $M = 2$ as an example. The size of the two-qubit Clifford group is 11520, however there are only 64 graphs on $2M$ qubits. Therefore, it follows that $\mathbf{L}_{T_A}(A) \leq 64$. This example already indicates that considering the possible graph states on the ancilla qubits might help to reduce the number of multi-qubit Cliffords relavant to check T-LMQC equivalence.

The next step in gate teleportion is to do Bell measurements in order to map U_M to $|G\rangle$. Given the set of all graph states in $\mathbf{L}_{T_A}(A)$, we will apply Bell measurements to study the effect on $|G\rangle$ of different graph states on the ancilla qubits. We will find that some of the graph states in $\mathbf{L}_{T_A}(A)$ will have a SQC-equivalent effect on $|G\rangle$. Remember that we are still using the naive approach described earlier in this section, so we first consider all multi-qubit nodes and then check for SQC-equivalence to $|G'\rangle$. Therefore, for two ancilla graph states which have a SQC-equivalent effect on $|G\rangle$, we only need to consider one of them in order to decide T-LMQC-equivalence of $|G\rangle$ and $|G'\rangle$. By removing the elements which have a duplicate effect (up to SQC-operations) from the Clifford group, we find the reduced Clifford set.



(a) A graph state $|G\rangle$ on 5 qubits with one 2 qubit node.

(b) A graph state $|G\rangle$ (below) and a graph state $|A\rangle$ (above) in the 2-Bell pair state. The rectangles denote measuring the two qubits in the Bell basis (Bell measurements).

Figure 5.2: An example for how a multi-qubit node in a graph state (in (a)) can be mapped to an ancilla state (in (b)). In order for the two to be equal, the multi-qubit operation on the ancilla state is mapped back to the original graph state via Bell measurements (the rectangles in (b)).

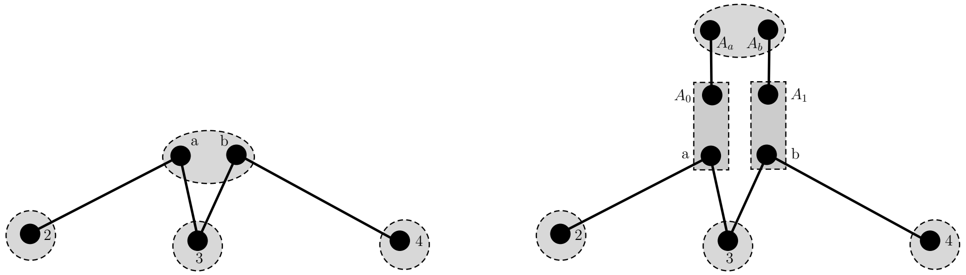
5.2. Formalizing the concept

In this section the approach described in the previous section will be formalized and an outline is given for the calculations that have to be done. Finally, the main result of this chapter is stated. In this section the number of qubits sharing a node is bounded by 2. In other words, we assume that there are only one- and two-qubit nodes in the graph state. However, in general this approach is not restricted by the number of qubits per node. In section 5.6 we discuss the general case without a bound on the number of qubits per node.

Let $|G\rangle$ be a graph state and let T_G be a partition of $V[G]$, the vertex set of G. Furthermore, assume that there exists a two-qubit node with two distinct qubits a, b, i.e. $\exists a, b : \exists t \in T_G : a, b \in t, a \neq b$. Let $T_G{}'$ be the partition T_G where $\{a, b\}$ is replaced by $\{a\}, \{b\}$. Let $|A\rangle$ be the 2-Bell pair graph state with every pair contributing a qubit to a multi-qubit node. I.e., $A = (V, E) = (\{A_0, A_1, A_a, A_b\}, \{(A_0, A_a), (A_1, A_b)\})$ and $T_A = (\{A_a, A_b\}, \{A_0\}, \{A_1\})$ is a partition of $V[A]$. Furthermore, let $U_{ab} \in \mathcal{C}_2$ and let G.T. $\left(G, \left|A^{U_{ab}}\right\rangle\right)$ denote applying operation U_{ab} to $|G\rangle$ via gate teleportation where $\left|A^{U_{ab}}\right\rangle = U_{ab}\,|A\rangle$. Note that when we write $U_{ab}\,|A\rangle$, U_{ab} is actually applied on A_a, A_b and not on a, b. Formally, we then have:

$$U_{ab}\left|G_{T_G}\right\rangle = \text{G.T.}\left(G_{T'_G}, \left|A^{U_{ab}}\right\rangle\right) \tag{5.1}$$

Note that in the r.h.s. of equation 5.1 we will drop the ket labels if the state is a graph state. So we write G instead of $|G\rangle$, but as $U_{ab}\,|A\rangle$ is not always a graph state we write $\left|A^{U_{ab}}\right\rangle$. For example, for a graph state on 5 vertices with qubits a,b sharing a node, figure 5.3(a) corresponds to the l.h.s. of equation 5.1 and figure 5.3(b) corresponds to the r.h.s. of the same equation. The rectangles correspond to Bell measurements of the two qubits inside, which are done in order to make the two graph states equal (for the same U_{ab} of course). Everything is in place now to formally discuss the concept of this chapter. When applying a two qubit gate directly to the graph state, $U_{ab}\,|G\rangle$, there are no intermediate steps that we can analyse. However, if we apply U_{ab} via gate teleportation there is a lot of redundancy on the ancilla state, namely the set of possible graph states on the ancilla qubits. Furthermore, we can analyse whether different ancilla states have a SQC-equivalent effect on $|G\rangle$. To state this more formally, for $U_{ab}, U'_{ab} \in \mathcal{C}_2$:



(a) Graph state $|G\rangle$ on the vertices $\{a, b, 2, 3, 4\}$ where a,b share a node.

(b) The same graph state as in (a) but now the multi-qubit operations on a,b are mapped onto the ancilla state. To end up with the same state as in (a), Bell measurements are done on qubits A_0, a and on A_1, b.

Figure 5.3: An example of equation 5.1, where we either apply a two-qubit gate directly on the graph state (a) or on an ancilla state (b). The Bell measurements, the rectangles in (b) correspond to Bell measurements on the two qubits.

$$\text{G.T.}\left(G, |A^{U_{ab}}\rangle\right) \overset{?}{\sim}_{SQC} \text{G.T.}\left(G, \left|A^{U'_{ab}}\right\rangle\right) \tag{5.2}$$

To be more specific, we first give a circuit view of applying a gate via gate teleportation in figure 5.4. When comparing this to the graphs we have seen before, the graph state in figure 5.3b corresponds to the graph state in figure 5.4 at L_1.

Let us focus on the redundancy of the states on the ancilla qubits by analysing the circuit in figure 5.4. The state $|A^U\rangle$ on the ancilla states at L_2 is in general not a graph state but a stabilizer state. From section 3.4 we know that any stabilizer state is SQC equivalent to a graph state. Therefore $|A^U\rangle$ can always be turned into a graph state $|A^U_{GS}\rangle$. We use subscript GS to denote that this is a graph state on the ancilla qubits originally in $|A\rangle$ and superscript U for the gate U_{ab} applied to $|A\rangle$. Note that U_{ab} fully specifies $|A^U_{GS}\rangle$. Furthermore, $|A^U_{GS}\rangle \in \mathbf{L}_{T_A}(A)$, i.e. $|A^U_{GS}\rangle$ is always T_A-LMQC equivalent to $|A\rangle$. Using algorithm 2 we know that there are only 24 graph states T_A-LMQC equivalent to $|A\rangle$. The fact that, for any two-qubit Clifford U, there are only 24 possible graph states on $|A^U_{GS}\rangle$ is the redundancy mentioned before.

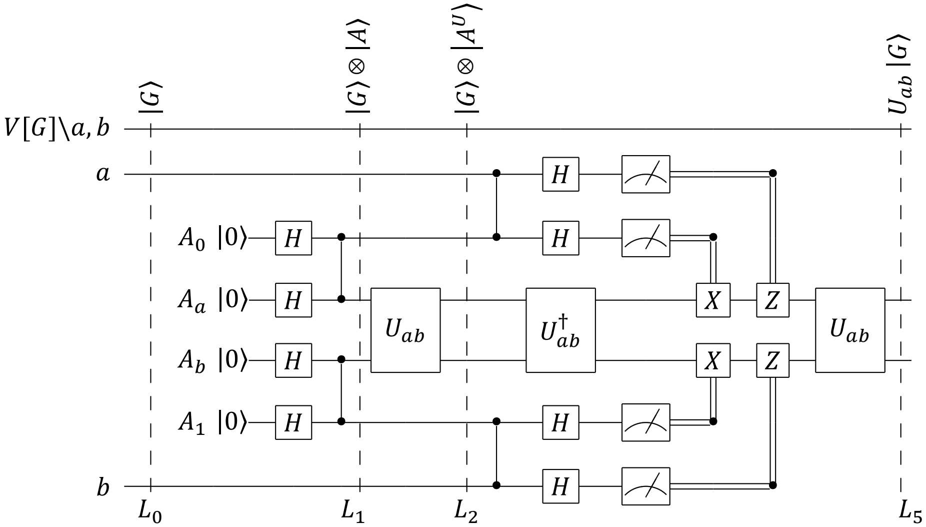


Figure 5.4: A circuit view of gate teleporting U_{ab} to $|G\rangle$. At the start of the circuit, at line L_0, we have the graph state $|G\rangle$. Next four ancilla qubits are added and prepared as the graph state $|A\rangle$, which makes the state at L_1 $|G\rangle \otimes |A\rangle$. U_{ab} is applied to ancilla qubits A_a,A_b and with $|A^U\rangle = U_{ab}\,|A\rangle$ the state after the two qubit gate, at L_2, is $|G\rangle \otimes |A^U\rangle$. After the Bell measurements the final state is $U_{ab}\,|G\rangle$ (at L_5).

However, just checking the 24 graph states in $\mathbf{L}_{T_A}(A)$ as an ancilla state in the gate teleportation circuit is not sufficient to consider T-LMQC equivalence of graph states. I.e.,:

$$\exists U_{ab} \in \mathcal{C}_2 : \forall A' \in \mathbf{L}_{T_A}(A) : U_{ab}\,|G\rangle \nsim_{SQC} \text{G.T.}(G, A') \tag{5.3}$$

What is missing here, is that the operations taking $|A^U\rangle$ to a graph state $|A^U_{GS}\rangle$ might affect the two-qubit gate applied to $|G\rangle$. Let $C_{\text{to graph}}$ be the single qubit Clifford operations such that $C_{\text{to graph}}\,|A^U\rangle = |A^U_{GS}\rangle$. We will now rewrite the definition of gate teleportation from equation 5.1 such that A^U_{GS} and $C_{\text{to graph}}$ is used:

$$\text{G.T.}(G, |A^U\rangle) = \text{G.T.}(G, A^U_{GS}, C_{\text{to graph}}) \tag{5.4}$$

By carefully analysing what the operations $C_{\text{to graph}}$ are and how they affect $|G\rangle$, we will find which combinations of A^U_{GS} and $C_{\text{to graph}}$ have a SQC-equivalent effect on $|G\rangle$. This leads us to the formal definition of the reduced Clifford set.

Definition 5.1. *Given a graph state $|G\rangle$ with a node with M qubits. Let $|A\rangle$ be the graph state corresponding to $A = (V, E) = (\{0, \dots, 2M-1\}, \{(0, 0+M), \dots, (M-1, M-1+M)\})$. For $U \in \mathcal{C}_M$, $|A^U\rangle = U\,|A\rangle$. Then $C_{\text{to graph}} \in \mathcal{C}_1^M$ is used to denote the SQC operations taking $|A^U\rangle$ to the corresponding graph state $|A^U_{GS}\rangle$. The reduced Clifford set $\mathcal{C}'_M$ on M qubits is a set of tuples $\tilde{C}$ where $\tilde{C} = (A^U_{GS}, C_{\text{to graph}})$ such that:*

$$\forall C \in \mathcal{C}_M : \exists \tilde{C} = \left(A^U_{GS}, C_{\text{to graph}}\right) \in \mathcal{C}'_M : G.T.\left(G, A^U_{GS}, C_{\text{to graph}}\right) \sim_{\text{SQC}} C\,|G\rangle \tag{5.5}$$

By $|\mathcal{C}_k{}'|$ we denote the number of elements in $\mathcal{C}_k{}'$. We are now ready to state the main result of this chapter.

Theorem 5.1. $\exists \mathcal{C}_2{}' : |\mathcal{C}_2{}'| = 40$

Section 5.4 provides lemmas that are used to prove theorem 5.1, the proof is given in section 5.3.

5.3. The reduced Clifford set for nodes with 2 qubits

In this section we proof theorem 5.1.

Proof of theorem 5.1. Let us start by providing an overview of the steps taken towards proving the theorem.

- Step 1: The circuit of figure 5.4 is further analysed to prepare for the other steps in the proof. Part of the previous sections is covered more formally. Furthermore, the operations are specified which take a stabilizer state to a graph state. This is covered in section 3.4.
- Step 2: The teleported gate can be substituted for a closely related gate which is chosen such that no SQC-operations are needed on qubits A_a, A_b to make the ancilla state a graph state. This is used for the first bound on the reduced Clifford set in eq. 5.12.
- Step 3: In the remaining steps (4-7) we will find that certain elements of the reduced Clifford set always result in SQC-equivalent stabilizer states. In this step, we simplify the gate teleportation circuit by assuming that the measurement outcomes of the Bell measurements are all 1 (referring to the $|0\rangle$ post-measurement state). Furthermore, new notation is introduced which will be used in the remainder of the proof.
- Step 4: Using lemma 5.3 it is noted that any $Z \in C_i$ never results in an unique (up to SQC) stabilizer state at L_5, in the sense that there always exists $C_i : Z \notin C_i$ with a SQC-equivalent effect. This leads to a thighter bound on the reduced Clifford set in equation 5.20.
- Step 5: Observe that single Hadamards (so $C_i = H$, but not $C_i = HS$) can be rewritten as a different ancilla graph state from $\mathbf{L}_T(A)$ with $C_i = \mathbb{1}$. This leads to a new bound in equation 5.24.

- Step 6: Note that stabilizer states for the corrections $C_i = HS$ are SQC-equivalent to one of the other C_i in the reduced Clifford set of equation 5.24 leading to the bound in equation 5.28.
- Step 7: Reduce the set of possible graph states on the ancilla qubits to 10 graph states. This is done by noting that by going to a graph state from a stabilizer state on the ancilla qubits, not just one of the 24 states in $\mathbf{L}_T(A)$ is reached, but actually one in a subset $(\mathbf{L}_T(A)')$ of 10 graphs is reached. To find this result, lemma 5.7 is used. This allows for an even tighter bound on the reduced Clifford set, given in equation 5.34.

Step 1
For this step we use the circuit given in figure 5.5. In this circuit $C_{\text{to graph}} = C_0 \otimes C_1 \otimes C_a \otimes C_b$ are chosen such that $|A^U_{GS}\rangle$ is always a graph state. In figure 5.5 the operations needed to transform $|A^U\rangle$ to a graph state are done between L_2 and L_3, and they are cancelled one step later between L_3 and L_4. In section 3.4 we discuss that one can always transform a stabilizer state to a graph state with only single qubit Cliffords.

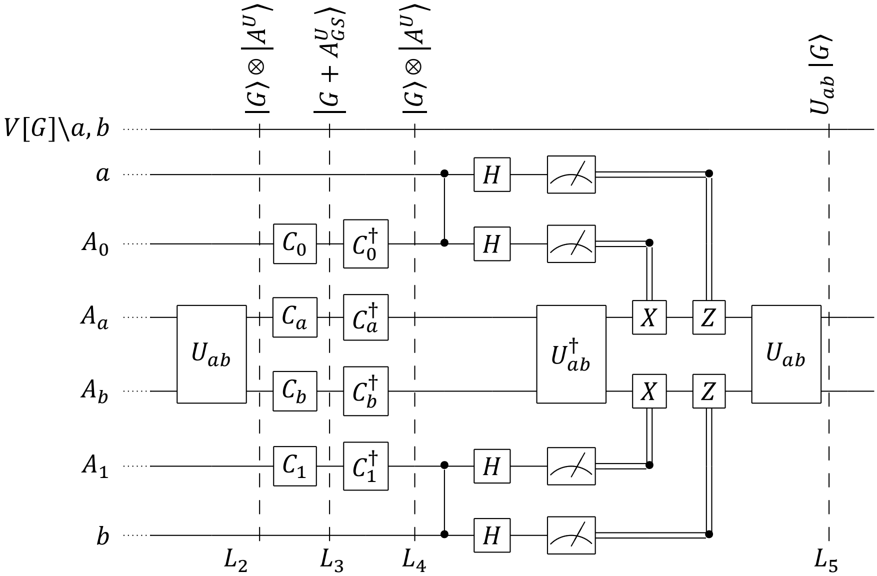


Figure 5.5: The gate teleportation without the initialization of the ancilla states (on A_0, A_1, A_a, A_b). C_i with $i \in \{0, 1, a, b\}$ are chosen such that the ancilla state is a graph state, $|A^U_{GS}\rangle$, and the operations are cancelled between L_3, L_4. At L_4 the state is the same as the state at L_2.

At L_3 the state on the ancilla qubits is a graph state, $|A^U_{GS}\rangle$. Furthermore, this graph state is reached by doing single qubit Cliffords on all four qubits and a two qubit Clifford between A_a and A_b on the original ancilla state $|A\rangle$. Then we know that $|A^U_{GS}\rangle \in \mathbf{L}_{T_A}(A)$, where $T_A = (\{A_a, A_b\}, \{A_0\}, \{A_1\})$ as before. I.e., $|A^U_{GS}\rangle$ is always a graph in the local multi-qubit Clifford equivalence class of $|A\rangle$ w.r.t. T_A. Using algorithm 2 one can find that there are only 24 such graph states ($|\mathbf{L}_{T_A}(A)| = 24$). So for every U_{ab} that is being teleported, $|A^U_{GS}\rangle$ is always one of the 24 graph states T_A-LMQC equivalent to $|A\rangle$. I.e.:

$$\forall U_{ab} \in \mathcal{C}_2 : \ \exists \left|A^U_{GS}\right\rangle \ \in \mathbf{L}_{T_A}(A) \tag{5.6}$$

At first sight it might seem that in order to check LMQC equivalence of graph states for a two qubit node, one only has to consider the 24 graph states in $\mathbf{L}_{T_A}(A)$ via the gate teleportation. However, this is not true as the operations making $|A^U_{GS}\rangle$ a graph state still have to be cancelled (between L_3 and L_4). If these operations are not cancelled, the two qubit gate U_{ab} applied just before L_5 might be a different Clifford gate than we originally teleported. One might try to commute the operations between L_3 and L_4 through the circuit to the right, so through the Bell measurements after L_5. In that case, only the 24 graph states in $\mathbf{L}_{T_A}(A)$ have to be teleported. However, commuting the operations from L_3-L_4 through till after L_5 does in general produce a two-qubit gate after L_5. As the idea is to check for SQC equivalence after gate teleportation, adding an extra two-qubit gate is not the way to go. Therefore, we will consider the circuit including cancellations, i.e. the operations between L_2 and L_3 are cancelled in the step between L_3 and L_4.

As discussed in section 3.4 the set of operations to transform any stabilizer state to a graph state consists of operations of the form $Z^z S^y H^x$ for $x, y, z \in \mathbb{Z}_2$. Therefore, we know that for $i \in \{0, 1, a, b\}$:

$$C_i \in C^0_{\text{to graph}} = \{\mathbb{1}, S, H, SH, Z, ZS, ZH, ZSH\} \tag{5.7}$$

Step 2

In this step we will change the gate U_{ab} teleported to another gate, which will bring us to the first non-trivial bound on the reduced Clifford set. First we will discuss what other gate is teleported instead of U_{ab}.

The operations on qubits A_a and A_b can be seen as operations belonging to the gate teleported, so instead of teleporting U_{ab}, the gate $U'_{ab} = (C_a \otimes C_b)U_{ab}$ is teleported. As C_a and C_b are SQC operations, it is clear that:

$$U_{ab} |G\rangle \sim_{\text{SQC}} U'_{ab} |G\rangle \tag{5.8}$$

This can also be stated in terms of gate teleported states:

$$\forall U_{ab} \in \mathcal{C}_2, \forall C_a, C_b \in C^0_{\text{to graph}} : \text{G.T.}(G, \left|A^{U_{ab}}\right\rangle) \sim_{SQC} \text{G.T.}(G, \left|A^{(C_a \otimes C_b)U_{ab}}\right\rangle) \tag{5.9}$$

Teleporting U'_{ab} allows us to write the circuit such that on qubits A_a and A_b no operations are needed to reach a graph state. This is shown in figure 5.6.

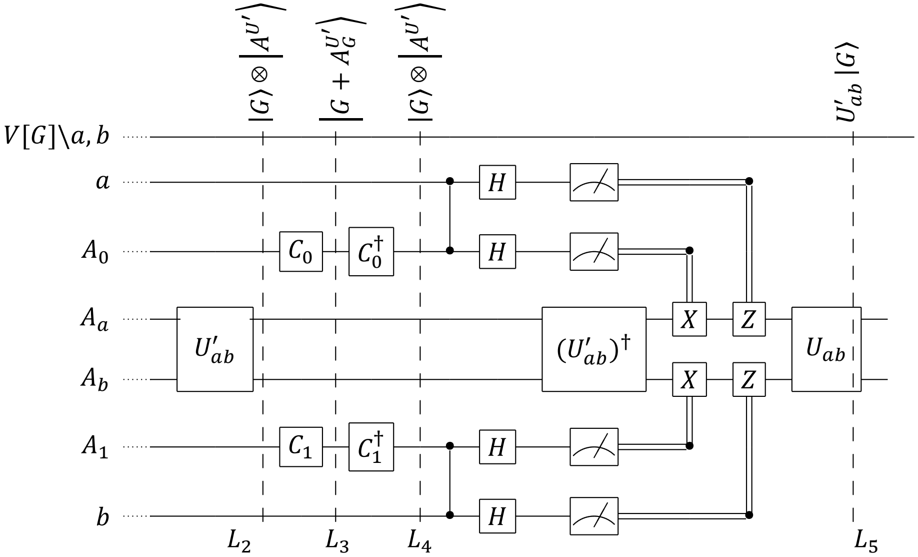


Figure 5.6: The gate teleportation circuit without initialization of the ancilla states. Furthermore, the teleported gate U_{ab} is substituted for $(C_a \otimes C_b)U_{ab} = U'_{ab}$. Note that the only remaining operations (C_i's) to reach a graph state on the ancilla qubits are on A_0 and A_1.

Next we will analyse the remaining operations between L_2 and L_4. We know from equation 5.7 that for $C_i \in \{C_0, C_1\}$, there are 8 possible gates to consider:

$$C^0_{\text{to graph}} = \{\mathbb{1}, H, S, SH, Z, ZH, ZS, ZSH\} \tag{5.10}$$

Therefore, the total number states at L_4 right now is 8×8 (for the operations on A_0 and A_1) times 24 (all possible graph states $\left|A^U_{GS}\right\rangle$) = 1536. The set of states can be labelled only by the graphs A^U_{GS} and the operations C_0, C_1 as follows:

$$C_2'^1 = \{(A^U_{GS}, C_0, C_1) | \forall A^U_{GS} \in \mathbf{L}_{T_A}(A), \forall C_0, C_1 \in C^0_{\text{to graph}}\} \tag{5.11}$$

This results in the first bound on the size of the reduced Clifford set:

$$\left|C_2'^1\right| = 1536 \tag{5.12}$$

Note that this is already almost x10 smaller than checking all elements of the two qubit Clifford group. The next step is to analyse the stabilizer state after the Bell measurements to see if some of these 1536 cases result in single qubit Clifford equivalent states.

Step 3

In order to do analyse the stabilizer state after the Bell measurements, we will use that the measurement outcome 0000 is always possible, i.e. the state $\left|0_a 0_b 0_{A_0} 0_{A_1}\right\rangle$ has non-zero probability of being the result of the four measurements. This follows from lemma 5.1. The circuit when only considering 0-outcome measurements is given in figure 5.7.

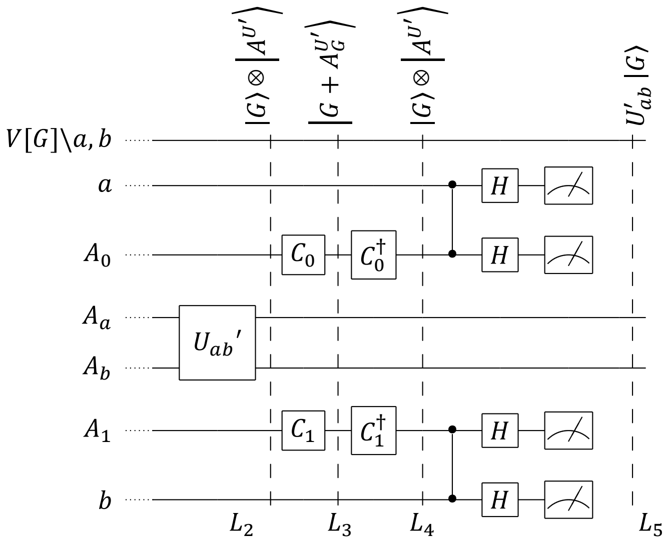


Figure 5.7: The gate teleportation circuit with the initialization part not incorporated in the figure. Furthermore, the measurement outcomes are assumed to be 0 in this circuit. Therefore there are no X, Z corrections after the measurements, and $U_{ab}^\dagger U_{ab}$ cancel.

We can now continue with analysing which of the 1536 cases result in SQC-equivalent stabilizer states after the Bell measurements. The Bell measurement part of the circuit corresponds to the following projective measurement: $\left|B^{i,j}\right\rangle = CZ_{i,j}H_iH_j\left|0_i0_j\right\rangle$. We can then describe the stabilizer state after the Bell measurements by starting with the state at L_3, a product state of the original graph state $|G\rangle$ and the ancilla graph state $\left|A_G^{U'}\right\rangle$: $\left|G + A_G^{U'}\right\rangle$. In the next step we consider the operations between L_3 and L_5: the cancelling of C_0 and C_1, which gives the state $C_0^\dagger C_1^\dagger\left|G + A_G^{U'}\right\rangle$, and the Bell measurements. When we denote the stabilizer state at L_5 by G.T.$^*(G, A_G^{U'}, C_0, C_1)$, it then follows that we can denote this state as:

$$U'_{ab}\,|G\rangle = \text{G.T.}^*(G, A_G^{U'}, C_0, C_1) = \left\langle B^{A_bA_1}\right|\left\langle B^{A_aA_0}\right| C_0^\dagger C_1^\dagger \left|G + A_G^{U'}\right\rangle \tag{5.13}$$

The astrix * is used to avoid confusing with the well-known gate teleportation circuit. Note that the operations between L_3 and L_5 correspond exactly to the operations in the definition in equation 5.13, this is of course by definition. Furthemore, note that this state depends on G, $A_G^{U'}$ and on C_0, C_1. The remainder of this proof is dedicated to see if there are different elements of $\mathcal{C}_2'^1$, $(A_{GS}^U, C_0, C_1), (\bar{A}_{GS}^U, \bar{C}_0, \bar{C}_1) \in \mathcal{C}_2'^1$, which always have a SQC-equivalent on $|G\rangle$ for every possible G, i.e.:

$$\forall G : \text{G.T.}^*(G, A_{GS}^{U'}, C_0, C_1) \sim_{\text{SQC}} \text{G.T.}^*(G, \bar{A}_{GS}^{U'}, \bar{C}_0, \bar{C}_1) \tag{5.14}$$

If we find such $(\bar{A}_{GS}^U, \bar{C}_0, \bar{C}_1)$, then these can be removed from $\mathcal{C}_2'^1$ without violating the definition of the reduced Clifford set.

There is one general constraint on G in this proof, which is the following. It is assumed that all local edges are removed before doing gate teleportation. I.e., inside every node all edges are removed. This can be trivially done by applying CZ gates before doing any of the gate teleportation as the qubits are in the same node. To use the notation currently used for the circuit, we have that $(a,b) \notin G$.

Step 4

In this step the Pauli-Z gates in

$$C^0_{\text{to graph}} = \{\mathbb{1}, H, S, SH, Z, ZH, ZS, ZSH\} \tag{5.15}$$

are considered and lemma 5.3 is used, which states that any single qubit Pauli before measurements results in multiple single qubit Pauli's after the measurement. Lemma 5.3 holds for any Pauli measurement, however it is not directly clear how this can be applied for Bell measurements. In this step we consider the effect of a Pauli-Z in $C^0_{\text{to graph}}$, and we see that there are four elements containing a Pauli-Z: Z, ZH, ZS, ZSH. As we will propagate Z through the Bell measurements and Z, S commute, we identify two cases: Z, ZH. Using the commutation relations from section 2.7 one finds:

$$Z_{A_0} CZ_{aA_0} H_a H_{A_0} = CZ_{aA_0} H_a H_{A_0} X_{A_0} Z_{A_0} H_{A_0} CZ_{aA_0} H_a H_{A_0} \quad = H_{A_0} CZ_{aA_0} H_a H_{A_0} Z_{A_0} X_a \tag{5.16}$$

The result is a Pauli-X or -Z on the ancilla qubit and a possible Pauli-X on the corresponding graph state qubit. These Pauli's will, according to lemma 5.3, result in a Pauli string after the measurements. To conclude, any $Z \in C_i$ can be propagated through the Bell measurements and the result will be a string of Pauli's. As this are only SQC operations after the multi-qubit operations, this Pauli-string will not affect the single qubit Clifford equivalence class. Therefore, we can disregard all elements of the reduced Clifford set with Z's in C_i for $C_i \in C^0_{\text{to graph}}$. More formal, with

$$C^1_{\text{to graph}} = \{\mathbb{1}, S, H, SH\} \tag{5.17}$$

we have:

$$\forall C_0, C_1 \in C^0_{\text{to graph}} = \{\mathbb{1}, H, S, SH, Z, ZH, ZS, ZSH\} : \exists \tilde{C}_0, \tilde{C}_1 \in C^1_{\text{to graph}} = \{\mathbb{1}, S, H, SH\} : \\ \text{G.T.}(G, A^U_{GS}, C_0, C_1) \sim_{SQC} \text{G.T.}(G, A^U_{GS}, \tilde{C}_0, \tilde{C}_1) \tag{5.18}$$

Therefore we can remove all elements with a Z from $\mathcal{C}'^1_2$, which results in $\mathcal{C}'^2_2$:

$$C'^2_2 = \{(A^U_{GS}, C_0, C_1) | \forall A^U_{GS} \in \mathbf{L}_{T_A}(A), \forall C_0, C_1 \in C^1_{\text{to graph}}\} \tag{5.19}$$

Which gives us a lower upper bound on the size of the reduced Clifford set compared to equation 5.12:

$$\left|C'^2_2\right| = 384 \tag{5.20}$$

Step 5

In this step we consider only Hadamard gates on A_0 or A_1, so $C_0 = H$ or $C_1 = H$. The effect of a single Hadamard on a graph state is not easy to see, but we know from section 3.1 that two Hadamards on adjacent vertices act as a pivot up to Pauli-Z on the common neighbours. Therefore, we will use an extra Hadamard on a connected but not measured vertex in $A_G^{U'}$, which is compensated by a Hadamard after the measurements. The possible Pauli-Z on the common neighbours is disregarded following the same reasoning as in Step 4. We will now discuss the calculations for $C_0 = H$ and $C_1 = \mathbb{1}$, the other cases are very similair and can be found in the appendix A.3.

$$\begin{aligned}
\text{G.T.}^* \left(G, A_G^{U'}, H, \mathbb{1}\right) =& \left\langle B_1^{bA_1}\right| \left\langle B_1^{aA_0}\right| H_{A_0} \left|G + A_G^{U'}\right\rangle \\
=& \left\langle B_1^{bA_1}\right| \left\langle B_1^{aA_0}\right| H_u H_u H_{A_0} \left|G + A_G^{U'}\right\rangle \\
=& H_u \left\langle B_1^{bA_1}\right| \left\langle B_1^{aA_0}\right| G + \rho_{A_0 u}(A_G^{U'})\right\rangle \\
=& H_u \text{G.T.}^* \left(G, \rho_{A_0 u}(A_G^{U'}), \mathbb{1}, \mathbb{1}\right) \\
\sim_{SQC}& \text{G.T.}^* \left(G, \rho_{A_0 u}(A_G^{U'}), \mathbb{1}, \mathbb{1}\right)
\end{aligned}$$

Where u is as either A_a or A_b, i.e. $u \in N_{A_0}(A_G^{U'}) \backslash A_1$. Lemma 5.6 proves that such an u always exists. Note that u is chosen as a vertex which is not measured. Otherwise, it might not commute with the measurements which is inconvenient. In the step from the second to the third line we use that u, A_0 are neighbours and that H_u commutes with the Bell measurements. The fourth line follows from the definition of G.T.*. To conclude, when considering all elements of the reduced Clifford set in equation 5.19, the elements with $C_0 = H$ or/and $C_1 = H$ will always be SQC-equivalent to some other elements with $C_0, C_1 = \mathbb{1}$. To state this formally:

$$\begin{gathered}
\forall C_0, C_1 \in \{\mathbb{1}, H, S, SH\} : \forall A_{GS}^U \in \mathbf{L}_{T_A}(A) : \exists \tilde{C}_0, \tilde{C}_1 \in \{\mathbb{1}, S, SH\} : \exists \tilde{A}_G^U \in \mathbf{L}_{T_A}(A) \\
\text{G.T.}(G, A_{GS}^U, C_0, C_1) \sim_{SQC} \text{G.T.}(G, \tilde{A}_G^U, \tilde{C}_0, \tilde{C}_1)
\end{gathered} \tag{5.21}$$

This new set for C_i is denoted by $C^2_{\text{to graph}}$:

$$C^2_{\text{to graph}} = \{\mathbb{1}, S, SH\} \tag{5.22}$$

Therefore, we can again remove some elements from $C_2'^2$ which leads to:

$$C_2'^3 = \{(A_{GS}^U, C_0, C_1) | \forall A_{GS}^U \in \mathbf{L}_{T_A}(A), \forall C_0, C_1 \in C^2_{\text{to graph}}\} \tag{5.23}$$

Which allows us to bound the size of the reduced Clifford group with a lower number (compared to eq. 5.12, 5.20):

$$\left|C_2'^3\right| = 24 \times 3 \times 3 = 216 \tag{5.24}$$

This number is already $\approx 50\times$ smaller than checking all elements of $\mathcal{C}_2$.

Step 6

In this step we will see that also $C_0 = SH$ and/or $C_1 = SH$ will lead to a state SQC-equivalent to a element of the reduced Clifford set with $C_0, C_1 \neq SH$, i.e.:

$$\begin{gathered}\forall C_0, C_1 \in \mathcal{C}^2_{\text{to graph}}, \forall A^U_{GS} \in \mathbf{L}_{T_A}(A) : \exists \tilde{C}_0, \tilde{C}_1 \in \mathcal{C}^3_{\text{to graph}}, \exists \tilde{A}^U_{GS} \in \mathbf{L}_{T_A}(A) : \\ \text{G.T.}^*(G, A^U_{GS}, C_0, C_1) \sim_{SQC} \text{G.T.}^*(G, \tilde{A}^U_{GS}, \tilde{C}_0, \tilde{C}_1)\end{gathered} \tag{5.25}$$

where

$$\mathcal{C}^3_{\text{to graph}} = \{\mathbb{1}, S\} \tag{5.26}$$

The calculations proving this statement are given in the appendix A.3. The remaining elements in the reduced Clifford set which might have a non SQC-equivalent effect are:

$$\mathcal{C}'^4_2 = \{(A^U_{GS}, C_0, C_1) | \forall A^U_{GS} \in \mathbf{L}_{T_A}(A), \forall C_0, C_1 \in \mathcal{C}^3_{\text{to graph}}\} \tag{5.27}$$

This results in a new bound on the size of the reduced Clifford set:

$$\mathcal{C}'^4_2 = 24 \times 2 \times 2 = 96 \tag{5.28}$$

Step 7

The last step of removing SQC-duplicate elements from the reduced Clifford set is based on a slightly different perspective. There is something else that we can do that we haven't discussed before. As before, C_i's just before L_3 make sure that $\left|A^{U'}_G\right\rangle$ is a graph state, and we know that $\left|A^{U'}_G\right\rangle \in \mathbf{L}_{T_A}(A)$. Furthermore, we know that $\left|\mathbf{L}_{T_A}(A)\right| = 24$, so there are 24 possible graph states $\left|A^{U'}_G\right\rangle$.

In this step we will add a few extra operations corresponding to local complementations on the qubits A_a, A_b. This is illustrated in figure 5.8. For convenience, we use the following notation for a sequence of local complementations $\tau_S = \tau_{S_0} \dots \tau_{S_{m-1}}$ where S is a string of length m with letters from $\{A_a, A_b\}$. For example, $S = A_a A_b A_a$ and therefore $\tau_S = \tau_{A_a} \tau_{A_b} \tau_{A_a}$. $A_i[\tau_S]$ is the single qubit Clifford operation on qubit A_i corresponding to the LC sequence τ_S. Local complementations map graph states to graph states, so the ancilla state at L'_3 is also a graph state: $\left|\tau_S\left(A^{U'}_G\right)\right\rangle$. However, by applying local complementations on A_a, A_b one has to apply a Clifford ($\sqrt{iZ}$) to the neighbours of A_a, A_b as well. Thus the Cliffords applied on A_0, A_1 due to local complementations on A_a, A_b are denoted by $A_0[\tau_S]$, $A_1[\tau_S]$. Just as before, we also cancel these operations, otherwise the propagate through the circuit and might result in an extra two-qubit gate on $|G\rangle$. Therefore, after L'_3 the daggered operations are applied. The LCs on A_a, A_b can be seen as part of U, just as before, we do not focus on these. For the new operations on A_0 and A_1 however, it is not directly clear how this affects the operations C_0,C_1.

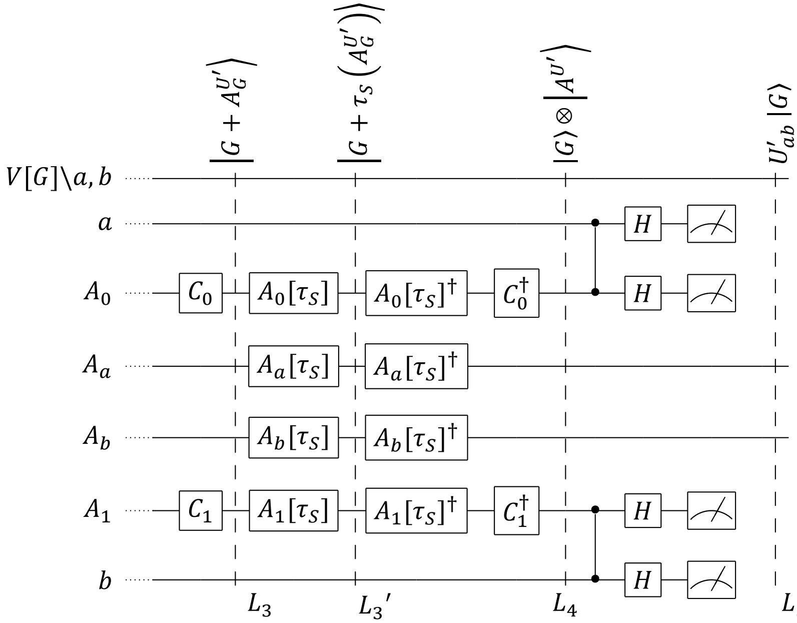


Figure 5.8: A part of the modified gate teleportation circuit without ancilla initialization and assuming that the measurement outcomes are all 0. The state of the qubits at L_3 is a product state of the graph state $|G\rangle$ and the graph state $\left|A_G^{U'}\right\rangle$. At L_3' the ancilla qubits are in the state $\left|\tau_S\left(A_G^{U'}\right)\right\rangle$.

For $j \in \{A_0, A_1\}$ we write the added operations, $A_j[\tau_S]$, in terms of the already possible operations on C_j. It follows that the operations on A_j applied between L_3 and L_3' are (sequences of) $S^\dagger$.

$$\frac{1}{\sqrt{i}}\sqrt{iZ} = S^\dagger \tag{5.29}$$

When we can now rewrite C_j to also incorporate $A_j[\tau_S]$ as:

$$C_j' = (S^\dagger)^l C_j \tag{5.30}$$

The set of possible operations for C_j' is then given by:

$$\mathcal{C}^4_{\text{to graph}} = \{(S^\dagger)^l C_j | l \in \mathbb{N}, C_j \in \mathcal{C}^0_{\text{to graph}}\} \tag{5.31}$$

Where $l \in \mathbb{N}$, indicating l local complementations on a neighbour of vertex j. On first hand, it might seem that $\mathcal{C}^4_{\text{to graph}}$ is a bigger set of operations compared to $\mathcal{C}^0_{\text{to graph}}$. However, in lemma 5.7 it is proven that $\forall C_j' \in \mathcal{C}^4_{\text{to graph}} : C_j' \in \mathcal{C}^0_{\text{to graph}}$. I.e., with the corrections in $\mathcal{C}^0_{\text{to graph}}$ every stabilizer state can be transformed to a graph state and local complementations can be done on A_a and A_b . Therefore, we don't have to consider all 24 graph states in $\mathbf{L}_{T_A}(A)$, but we only have to consider the graph states which are not equivalent under local complementations on A_a, A_b. This set of graph states is called $\mathbf{L}_{T_A}(A)'$ and an example of this set is shown in figure 5.9, where $\left|\mathbf{L}_{T_A}(A)'\right| = 10$. Again, we will now state this formally:

$$\begin{gathered}\forall C_0, C_1 \in \mathcal{C}^3_{\text{to graph}}, \forall A^U_{GS} \in \mathbf{L}_{T_A}(A) : \exists \tilde{C}_0, \tilde{C}_1 \in \mathcal{C}^3_{\text{to graph}}, \exists \tilde{A}^U_{GS} \in \mathbf{L}_{T_A}(A)' : \\ \text{G.T.}^*(G, A^U_{GS}, C_0, C_1) \sim_{SQC} \text{G.T.}^*(G, \tilde{A}^U_{GS}, \tilde{C}_0, \tilde{C}_1)\end{gathered} \tag{5.32}$$

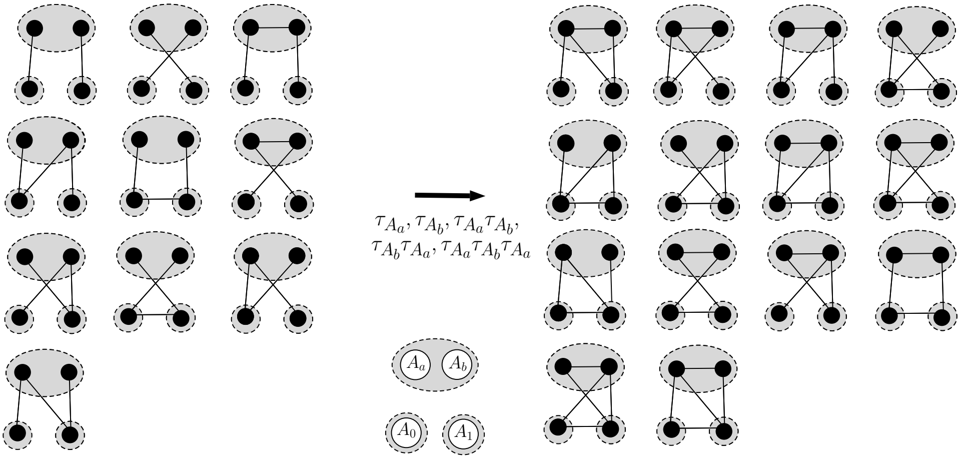

$\tau_{A_a}, \tau_{A_b}, \tau_{A_a}\tau_{A_b},$
$\tau_{A_b}\tau_{A_a}, \tau_{A_a}\tau_{A_b}\tau_{A_a}$
A_a
A_b
A_0
A_1

Figure 5.9: All 24 graph states LMQC equivalent to two Bell pairs where both pairs contribute one qubit to a shared node. The r.h.s. graphs can be reached from l.h.s. graphs by only doing local complementations on A_a and A_b.

Now we can make the final reduction on the reduced Clifford set. By only considering the ancilla states in $\mathbf{L}_{T_A}(A)'$ we end up with the reduced Clifford set:

$$C_2'^5 = \{(A^U_{GS}, C_0, C_1) | \forall A^U_{GS} \in \mathbf{L}_{T_A}(A)', \forall C_0, C_1 \in \mathcal{C}^3_{\text{to graph}}\} \tag{5.33}$$

Which lead to the final result of this proof:

$$\left|C_2'^5\right| = 2 \times 2 \times 10 = 40 \tag{5.34}$$

This concludes the proof. □

5.4. Lemmas for the proof in 5.3

This section is dedicated to proving multiple lemmas which are used in the proof of theorem 5.1. Some will be used explicitly in the proof, others will be used in the calculations in appendix A.3. The results of section 5.4.2 might be of independent interest, whereas the results of the other subsections are very specific to our needs in proving theorem 5.1.

5.4.1. We might always measure zeros

In this section we prove lemma 5.1, which is used in the proof of theorem 5.1.

Lemma 5.1. *In the gate teleportation circuit in figure 5.4 the probability of collapsing the state, by the measurements, of the four measured qubits to $|0000\rangle$ is bigger than 0.*

Proof. We start with figure 5.4, where we shift all operations on qubits A_a,A_b such that in the circuit they appear after the measurements of the other qubits. Then we get the circuit given in figure 5.10. Note that this shifting does not change the result of the circuit in anyway. Note that after the ancilla initialization, before the second vertical line, we have a product state of two graph states. The two CZ gates connect the two graph states and the Hadamards act as a pivot. As a, A_0 are not from different neighbourhoods of (b, A_1), because of the assumption that $(a, b) \notin G$, and using lemma 5.4 we find that $|\tilde{G}\rangle$ is indeed a graph state. To continue, we use the following proposition:

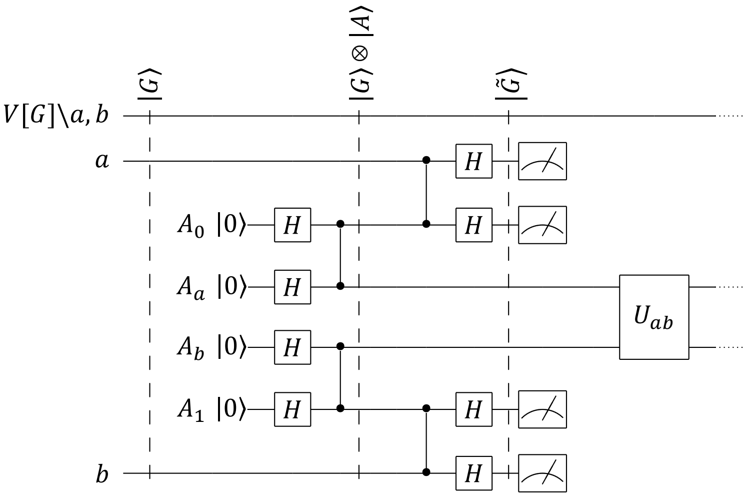


Figure 5.10: The first part of the gate teleportation circuit from figure 5.4. All operations on A_a, A_b after the initialization as a graph state are delayed till after the measurements of the other four qubits. The state just before the measurements is the graph state $|\tilde{G}\rangle = |\rho_{a,A_0}\rho_{b,A_1}(G + A + (a, A_0) + (b, A_1))\rangle$.

Lemma 5.2 (proposition 8 in [18]). *Let $A \subseteq V$ be a subset of vertices for a graph $G = (V, E)$ and $B = V\backslash A$ the corresponding complement of A in V. The reduced state $\rho_G^A := tr_B(|B\rangle\langle B|)$ is given by*

$$\rho_G^A = \frac{1}{2^{|A|}} \sum_{\sigma \in \mathcal{S}_A} \sigma \ , \tag{5.35}$$

where $S_A := \{\sigma \in \mathcal{S} : supp(\sigma) \subseteq A\}$ *denotes the subgroup of stabilizer elements* $\sigma \in \mathcal{S}$ *for* $|G\rangle$ *with support on the set of vertices within* A. ρ_G^A *is up to some factor a projection,i.e.,*

$$(\rho_G^A)^2 = \frac{|\mathcal{S}_A|}{2^{|A|}} \rho_G^A \ . \tag{5.36}$$

It projects onto the subspace in $\mathbf{H}^A$ *spanned by the vectors*

$$|\Gamma' B'\rangle_{G[A]} = Z^{Gamma' B'} \left|G[A]\right\rangle \quad (B' \subset B) \ , \tag{5.37}$$

where $G[A] = G \backslash B$ *is the subgraph of* G *induced by* A *and* $\Gamma' := \Gamma^{AB}$ *denotes the* $|A| \times |B|$*-off-diagonal sub-matrix of the adjacency matrix* Γ *for* G *that represents the edges between* A *and* B. *In this basis,* ρ_G^A *can be written as*

$$\rho_G^A = \frac{1}{2^{|B|}} \sum_{B' \subseteq B} |\Gamma' B'\rangle_{G[A]} \langle \Gamma' B'| \tag{5.38}$$

5

In our case, $A = \{a, A_0, b, A_1\}$. From the definition of graph states we know that $|0000\rangle \in \left|G[A]\right\rangle$. Furthermore, the Pauli-Z in equation 5.37 does not affect the $|0000\rangle$ state. Therefore, it follows from equation 5.38 that $|0000\rangle\langle 0000| \in \rho_G^A$. Therefore, the resulting state of measuring the four qubits can always be $|0000\rangle$, i.e. the probability of this to happen is bigger than 0. □

5.4.2. Pauli's and Pivots

In this subsection we cover a lemma on Pauli-measurements and the relation between a sequence of Hadamards and pivot operations. We will start by rewriting the single Pauli measurement rules from section 3.1 such that it is clear that for a given basis, the two possible post-measurement states differ by only Pauli's. This might seem trivial at first, but it will be very convenient. Afterwards, we use this result to derive that a Pauli operation acting on the graph state before measurement is equal to a Pauli on multiple qubits after the measurement. We also discuss when 4 Hadamard gates correspond to 2 pivots on a graph state in section 5.4.2. Finally, we derive measurement rules for measuring two qubits in a graph state in section 5.4.2.

Rewriting Pauli measurement rules for graph states

Let $|G\rangle$ be a graph state on N qubits and $v \in V[G]$. Lets use the following notation for projectors:

$$P_{i,\pm}^a = |i, m\rangle\langle i, m| \tag{5.39}$$

where $i \in \{x, y, z\}$ and $m \in GF(2)$ such that $m = \begin{cases} 0 & \text{for } P_{i,+}^a \\ 1 & \text{for } P_{i,-}^a \end{cases}$. Furthermore, inline square brackets following an operation are sometimes used for convenience:

$X[V] = \prod_{i\in V} X_i$. An integer m can be used as an exponent: $X[V]^m = \prod_{i\in V} X_i^m$.

Pauli-Z

In the expression for Z measurements it is directly clear that a different m only affects Pauli's. To be precise, m only appears in Pauli related terms.

$$|z,m\rangle\langle z,m|_v |G\rangle = \frac{1}{\sqrt{2}} |z,m\rangle_v Z[N_v(G)]^m |G\backslash v\rangle \tag{5.40}$$

Pauli-Y

For the Pauli-Y measurements we split a Clifford using $\sqrt{-iZ}Z = \sqrt{iZ}$ in order to see the effect of different measurement outcomes m.

$$\begin{aligned} |y,m\rangle\langle y,m|_v |G\rangle &= \frac{1}{\sqrt{2}} |m\rangle_v \sqrt{(-1)^{1-m} iZ}[N_v(G)] \left|\tau_v(G)\backslash v\right\rangle \\ &= \frac{1}{\sqrt{2}} |m\rangle_v \sqrt{-iZ}[N_v(G)]Z[N_v(G)]^m \left|\tau_v(G)\backslash v\right\rangle \end{aligned} \tag{5.41}$$

Pauli-X

For Pauli-X we apply the same method as for Pauli-Y, although it is a bit more involved now. The main step is to use $\sqrt{iY}(-iY) = \sqrt{-iY}$ to remove the m term from the square root.

$$\begin{aligned} |x,m\rangle\langle x,m|_v |G\rangle &= \begin{cases} \frac{1}{\sqrt{2}} |x,m\rangle_v \sqrt{+iY_u} Z[N_v\backslash(N_u \cup u)] \left|\rho_{uv}(G)\backslash v\right\rangle & \text{if } m = 0 \\ \frac{1}{\sqrt{2}} |x,m\rangle_v \sqrt{-iY_u} Z[N_u\backslash(N_v \cup v)] \left|\rho_{uv}(G)\backslash v\right\rangle & \text{if } m = 1 \end{cases} \\ &= \frac{1}{\sqrt{2}} |x,m\rangle_v \sqrt{iY_u} \left(-iY_u\right)^m Z[N_v\backslash(N_u \cup u)]^{m+1} Z[N_u\backslash(N_v \cup v)]^m \left|\rho_{uv}(G)\backslash v\right\rangle \end{aligned} \tag{5.42}$$

and $u \in N_v(G)$.

A Pauli before leads to Pauli's after measurements

Here we will prove that Pauli's acting on a graph state before measuring, are equivalent to other Pauli's acting on the graph state after measuring. , we have the following lemma.

Lemma 5.3. *Let P_v be a single qubit Pauli, i.e. $P_v \in \mathcal{P}^1$ and let $\tilde{P} \in \mathcal{P}^N$. Furthermore, let $i \in x, y, z$ and $m \in GF(2)$. Using the notation for projectors from equation 5.39, we then have:*

$$|i,m\rangle\langle i,m|_v P_v |G\rangle = \tilde{P}_v |i,m\rangle\langle i,m|_v |G\rangle$$

Proof. This proof will consider every measurement basis m_v^i separately. The main idea for every calculation is to see that any Pauli either acts trivially or flips the measurement basis to the other eigenvector. This flip will only contribute an extra Pauli string which we can pull out to find the original measurement projectors.

- When $P_v = i$, the projector and the P_v commute. Therefore, the Pauli acts trivially and we find that $P_v = \tilde{P}_v$.
- Otherwise, if $P_v \neq i$, then:

$$|i,m\rangle\langle i,m|\, P_v\, |G\rangle = |i,m\rangle\, \langle i,m+1|G\rangle \tag{5.43}$$

From the previous section, equations 5.40-5.42, we know that this is $\tilde{P}\, |i,m\rangle\langle i,m|\, |G\rangle$. To be precise, we have:

$$\tilde{P}_v = \begin{cases} Z[N_v(G)] & \text{if } i = Z \\ (iZ)[N_v(G)] & \text{if } i = Y \\ (-iY_v)\, Z[N_v \backslash (N_u \cup u)] Z[N_u \backslash (N_v \cup v)] & \text{if } i = Z \end{cases} \tag{5.44}$$

This shows that every Pauli applied on a graph state before measuring, is equivalent to a Pauli(s) after measuring. □

Multiple Hadamards and multiple pivots

In this section it is discussed how multiple Hadamards act on a graph state. The same notation is used as in section 3.2. Given a graph (state) G ($|G\rangle$) where $a, b, c, d \in V[G]$ are four distinct vertices. To denote the sets of neighbours in G of c and d, c but not d and d but not c as $N_{cd} = N_c \cap N_d$, $N_{c\backslash d} = N_c \backslash N_d$ and $N_{d\backslash c} = N_d \backslash N_c$. We denote $N^G(c,d) = (N_{cd}, N_{c\backslash d}, N_{d\backslash c})$ as the neighbourhood set of (c,d).

Definition 5.2. *Two vertices a and b are from a different neighbourhood of (c,d) if $a \in N_i$ and $b \in N_j$ for $i \neq j$ and $i, j \in \{cd, c\backslash d, d\backslash c\}$.*

Definition 5.3. *(a,b) is a neighbourhood connector of (c,d) if $(a,b) \in G$ and a and b are from a different neighbourhood of (c,d).*

This leads to the following lemma:

Lemma 5.4. *$PH_aH_bH_cH_d\,|G\rangle$ is equal to $|\rho_{ab}\rho_{cd}(G)\rangle$ if one of the following statements is true.*

- *$(a,b) \in G$ and a and b are not from different neighbourhoods of (c,d).*

- *$(a,b) \notin G$ and a and b are from different neighbourhoods of (c,d).*

where $P \in \mathcal{P}_V$.

Proof. From lemma 3.1 it is known that if $(a,b) \in \rho_{cd}(G)$ then $H_a H_b \left|\rho_{cd}(G)\right\rangle = \left|\rho_{ab}\rho_{cd}(G)\right\rangle$ up to a global phase and a Pauli-Z. Furthermore, since only simple graphs are considered, $(a,b) \in G$ or $(a,b) \notin G$. Lets cover these two cases separately:

- $(a,b) \in G$: In order for $H_a H_b$ to act as a pivot, it must be that $(a,b) \in \rho_{cd}(G)$. As $(a,b) \in G$ before ρ_{cd}, it follows that ρ_{cd} must not complement (a,b). This is true if a and b are not from different neighbourhoods.
- $(a,b) \notin G$: Then (a,b) should be complemented by a pivot on (c,d), so a and b should be from different neighbourhoods of (c,d).

We haven't considered the possible Pauli-Z as a result of a pivot. From lemma 3.1 we know that $H_c H_d$ act as a pivot on G up to $Z[N^G_{cd}]$. This Pauli term can be commuted with the remaining $H_a H_b$ to find $Z[N_{cd}\backslash\{a,b\}]X[N_{cd} \cap \{a,b\}]$. For applying $H_a H_b$ on $\left|\rho_{cd}(G)\right\rangle$ there is also an extra Z term. By combining this, we find that:

$$P = Z[N_{cd}\backslash\{a,b\}]X[N_{cd} \cap \{a,b\}]Z[N_{ab}^{\rho_{cd}(G)}] \tag{5.45}$$

□

To our knowledge, it is not known what the effect is of two Hadamards acting on non-adjacent vertices of a graph state. In general, the resulting state is not a graph state. However, just as with Hadamards on adjacent vertices in lemma 3.1, it might be that there is a general rule using the stabilizers of the graph state. If such a rule exists, it might be that Hadamards on non-adjacent vertices transform graph states to graph states up to some Pauli operators.

The next lemma is used to simplify the calculations in the appendix A.3. Let $T_A = (\{A_a, A_b\}, \{A_0\}, \{A_1\})$, $|A'\rangle \in \mathbf{L}_{T_A}(Bell_2)$ and $|G\rangle$ a graph state with more than 1 qubit. Let $a, b \in V[G]$ be two distinct vertices in G. We assume that $(a,b) \notin G$, which is the same assumption as made in the proof of theorem 5.1. This is visualized in figure 5.11.

Lemma 5.5. *Let $G' = \rho_{aA_0}\rho_{bA_1}(G + A' + (a, A_0) + (b, A_1))$. Then:*

$$(a,b) \notin G \iff (A_0, A_1) \notin G' \tag{5.46}$$

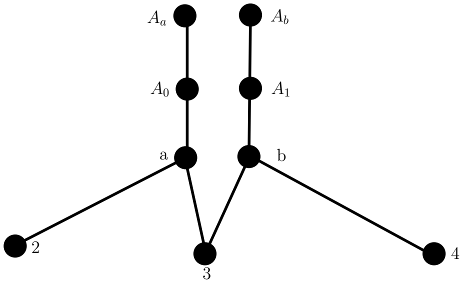


Figure 5.11: An example used in lemma 5.5. The lemma states that after pivots on A_0, a and A_1, b, the edge (A_0, A_1) is not present if $(a, b) \notin G$.

Proof. As $(a, b) \notin G$, we know that (a, A_0) is never a neighbourhood connector of (b, A_1). Therefore, ρ_{bA_1} does not flip (a, A_0). From section 3.2 it is known how a pivot changes a graph state. We first apply ρ_{bA_1}, which exchanges the neighbours of b and A_1, from which we find that $(a, b) \in G \iff (a, A_1) \in \rho_{bA_1}(G + A' + (a, A_0) + (b, A_1))$. The pivot on a and A_0 exchanges the neighbours of a and A_0, so that we have $(a, A_1) \in \rho_{bA_1}(G + A' + (a, A_0) + (b, A_1)) \iff (A_0, A_1) \in \rho_{aA_0}\rho_{bA_1}(G + A' + (a, A_0) + (b, A_1))$. Combining this with the earlier statement proves the lemma. □

Graphical rules for Pauli measurements on two qubits in a graph state

In this subsection we will discuss what happens to a graph state $|G\rangle$ on N qubits when two qubits a and b are measured. The results of this section are used in the calculations in appendix A.3 which are needed in the proof of theorem 5.1. However, this subject might also be of interest in general. We will distinguish between Pauli-X,Y and -Z measurements. In order to do this we use the single Pauli measurement rules introduced in subsection 3.2. Square brackets in a superscript are used as follows: $x^{[S]} = x$ if $S \in \{1, True\}$, $x^{[S]} = 1$ otherwise. We will first discuss measuring one or more qubits in the Z basis, then two Y and two X measurements and finally an X and Y measurement. We will consider the following state:

$$|i_a, m_a\rangle\langle i_a, m_a| \, |i_b, m_b\rangle\langle i_b, m_b| \, |G\rangle \tag{5.47}$$

where $i_a, i_b \in \{x, y, z\}$ and $m_a, m_b \in GF(2)$. This notation is introduced in section 3.1

One or more Pauli-Z measurements

Here we discuss what happens to a graph state when one qubit (a) is measured in a Pauli basis and the other (b) is measured in the Pauli-Z basis. Therefore, we are interested in the projectors in equation 5.47 with $i_a \in x, y, z$ and $i_b = z$:

$$\begin{aligned}|i_a, m_a\rangle\langle i_a, m_a|\, |z_b, m_b\rangle\langle z_b, m_b|\, |G\rangle = &|i_a, m_a\rangle\langle i_a, m_a|\, |z_b, m_b\rangle\, Z\,[N_b(G)]^{m_b}\, |G\backslash b\rangle \\ =&Z\,[N_b(G)]^{m_b}\, \tilde{P}_a\, |i_a, m_a\rangle\, |z_b, m_b\rangle\, \langle i_a, m_a|G\backslash b\rangle \\ \sim_{SQC}& |i_a, m_a\rangle\, |z_b, m_b\rangle\, \langle i_a, m_a|G\backslash b\rangle\end{aligned}$$

From the first to the second line we use lemma 5.3 to shift the potential Z's on $N_b(G)$ to after the measurements, where we have an extra $\tilde{P}_a \in \mathcal{P}_1^N$ to incorporate the shifting of Paulis. Note that $\tilde{P}_a$ depends on i_a, m_a and m_b. In the last line we disregard all SQC operations after the measurement to find the resulting graph state.

Two Pauli-Y measurements
For two Y-Pauli measurements we do the same as for one or more Z's, however the operations become a bit more involved. However, if $(a, b) \notin G$ the corrections from measuring one cant change the measurement basis of the other. For $(a, b) \in G$ we have:

$$\begin{aligned}&|y_a, m_a\rangle\langle y_a, m_a|\, |y_b, m_b\rangle\langle y_b, m_b|\, |G\rangle = \\ &\quad= |y_a, m_a\rangle\langle y_a, m_a|\, |y_b, m_b\rangle \sqrt{(-1)^{1-m_b} iZ}[N_b(G)]\, |\tau_b(G)\backslash b\rangle \\ &\quad=\sqrt{(-1)^{1-m_b} iZ}[N_b(G)\backslash a]\, |y_a, m_a\rangle\, |y_b, m_b\rangle\, \langle y_a, m_a|\, \sqrt{(-1)^{1-m_b} iZ_a}\, |\tau_b(G)\backslash b\rangle \\ &\quad=\sqrt{(-1)^{1-m_b} iZ}[N_b(G)\backslash a]\, |y_a, m_a\rangle\, |y_b, m_b\rangle\, \langle x_a, m_a + m_b|\tau_b(G)\backslash b\rangle \\ &\quad=\sqrt{(-1)^{1-m_b} iZ}[N_b(G)\backslash a]\, |y_a, m_a\rangle\, |y_b, m_b\rangle\, U^u_{x, m_a+m_b}\, |\tau_a\tau_u\tau_a\tau_b(G)\backslash a, b\rangle \\ &\sim_{SQC} |y_a, m_a\rangle\, |y_b, m_b\rangle\, |\tau_a\tau_u\tau_a\tau_b(G)\backslash\ a, b\rangle\end{aligned}$$

Where $u \in N_a(\tau_b(G))\backslash b$ if such a u exists, otherwise a is already in an eigenstate of Pauli-X. U_x^m is the correction corresponding to a X measurement with outcome m. To the third line we first use the single qubit measurement rule from equation 5.41 and keep only the operation on qubit a. Then using $\langle y, m_a|\sqrt{(-1)^{1-m_b} iZ} = \langle x, m_a + m_b|$ we find that a is now measured in the X basis. Using the single qubit Pauli measurement rules we find the result. By assuming that $|N_a^G| > 1$ if $(a, b) \in G$, we find:

$$|y_a, m_a\rangle\langle y_a, m_a|\, |y_b, m_b\rangle\langle y_b, m_b|\, |G\rangle \sim_{SQC} |y_a, m_a\rangle\, |y_b, m_b\rangle\, |(\tau_a\tau_u)^{[(a,b)\in G]}\tau_a\tau_b(G)\backslash\ a, b\rangle \tag{5.48}$$

If the assumption is not valid, a is disconnected from the rest of the graph after removing b. Therefore, the X measurement will not change the state. The resulting state is SQC equivalent to $|y_a, m_a\rangle\, |y_b, m_b\rangle\, |\tau_b(G)\backslash\ a, b\rangle$.

Two Pauli-X measurements
Both qubits are measured in the X basis so we have equation 5.47 with $i_a, i_b = x_a, x_b$. When $(a, b) \notin G$ the first measurement does not change the second (up to

Pauli's afterwards depending on u). Otherwise, if $(a,b) \in G$, we use the fact that $\langle x,m|\, H = \langle z,m|$, which leads to:

$$\begin{aligned} &|x_a,m_a\rangle\langle x_a,m_a|\,|x_b,m_b\rangle\langle x_b,m_b|\,|G\rangle = \\ &\qquad = |z_a,m_a\rangle\langle z_a,m_a|\,|z_b,m_b\rangle\langle z_b,m_b|\,H_aH_b\,|G\rangle = \\ &\qquad = |z_a,m_a\rangle\langle z_a,m_a|\,|z_b,m_b\rangle\langle z_b,m_b|\,Z[N_a^G \cap N_b^G]\,|\rho_{ab}(G)\rangle = \\ &\quad \sim_{SQC} |x_a,m_a\rangle\,|x_b,m_b\rangle\,|\rho_{a,b}(G) \setminus a,b\rangle \end{aligned}$$

When we also incorporate $(a,b) \notin G$, this leads to the general rule:

$$|x_a,m_a\rangle\langle x_a,m_a|\,|x_b,m_b\rangle\langle x_b,m_b|\,|G\rangle \sim_{SQC} \begin{cases} |x_a,m_a\rangle\,|x_b,m_b\rangle\,|\rho_{a,b}(G) \setminus a,b\rangle & \text{if } (a,b) \in G \\ |x_a,m_a\rangle\,|x_b,m_b\rangle\,|\rho_{a,v}\rho_{b,u}(G)\setminus a,b\rangle & \text{if } (a,b) \notin G \end{cases} \tag{5.49}$$

Pauli-X and a Pauli-Y measurement

We first consider the Y measurement, which flips the X measurement to the Y basis if $(a,b) \in G$. With flip, we refer to $\sqrt{(-1)^{1-m_b}iZ_a}\,|x_a,m_a\rangle = |y_a,m_a+m_b\rangle$. Otherwise, if $(a,b) \notin G$, the two graph operations do not affect each other. This results in a measurement rule for measuring two qubits, one in X and one in Y basis:

$$|x_a,m_a\rangle\langle x_a,m_a|\,|y_b,m_b\rangle\langle y_b,m_b|\,|G\rangle \sim_{SQC} |x_a,m_a\rangle\,|y_b,m_b\rangle\,|(\tau_a\tau_u)^{[(a,b)\notin G]}\tau_a\tau_b(G)\setminus a,b\rangle \tag{5.50}$$

Where $u \in N_a(\tau_b(G)\backslash b)$.

5.4.3. Specialized lemmas for section 5.3

In this section we cover lemmas which are needed for the proof of theorem 5.1 and which are probably not of interested outside the scope of this proof.

Lemma 5.6. *Let* $Bell_2^{T_A} = (V,E) = (\{A_a,A_b,A_0,A_1\},\{(A_0,A_a),(A_1,A_b)\})$ *with* $T = (\{A_a,A_b\},\{A_0\},\{A_1\})$ *then:*

$$\forall\,|G\rangle \in \mathbf{L}_T(Bell_2), \forall i \in \{0,1\} : (A_i,A_a) \in G \vee (A_i,A_b) \in G \tag{5.51}$$

$Bell_2^{T_A}$ is shown in figure 5.12(a). To state this lemma in words, for all graph states T-LMQC equivalent to $|Bell_2\rangle$, A_0 and A_1 are always connected to A_a, A_b or to A_a and A_b.

Proof. Using algorithm 2 one can find all graph states T-LMQC to $Bell_2$. An implementation in SAGE to do this can be found on our GitHub repository [12]. The result is the set of graphs in figure 5.12. For every graph in figure 5.12 the lemma is true, which concludes the proof. □

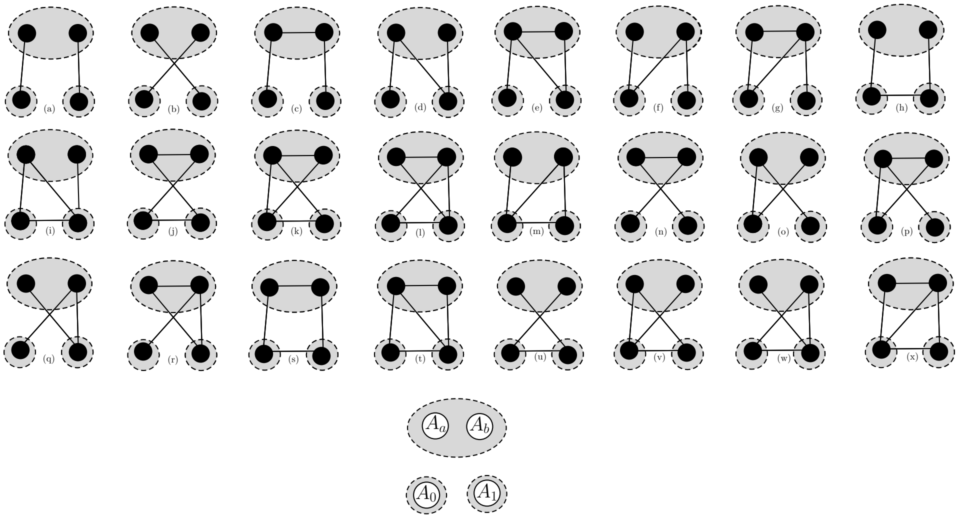


Figure 5.12: All 24 graph states LMQC equivalent to two Bell pairs where both pairs contribute one qubit to a shared node. The labelling is shown below the graphs.

Lemma 5.7. *For* $x, y, z, i, j, k \in \mathbb{Z}_2$ *and* $l \in \mathbb{N}$

$$\forall i, j, k, l\ \exists x, y, z : H^x (S^\dagger)^y Z^z = H^i (S^\dagger)^j Z^k \sqrt{-iZ}^l \tag{5.52}$$

Proof. In this proof we will write rotations around the Z-axis by specifying the rotated angle in radians, we will therefore use a letter $\mathcal{Z}$ as follows: $Z = \mathcal{Z}(\pi)$, $S = \mathcal{Z}(\pi/2)$, $S^\dagger = \mathcal{Z}(-\pi/2)$. We first note that $\sqrt{-iZ}_l = \frac{(1-i)}{\sqrt{2}} S_l = \frac{(1-i)}{\sqrt{2}} \mathcal{Z}(90l)$ and $S = (S^\dagger)^3$. Up to a global phase we can then write the r.h.s. of equation 5.52 as:

$$H^i \mathcal{Z}\left(k\pi + (l-j)\frac{\pi}{2}\right)$$

Note that $\mathcal{Z}(2\pi) = \mathbb{1}$, from which we conclude that the argument of $\mathcal{Z}$ can be simplified to four cases.

$$\mathcal{Z}\left(k\pi + (l-j)\frac{\pi}{2}\right) = \begin{cases} \mathbb{1} & \text{if } k\pi + (l-j)\frac{\pi}{2} \mod 2\pi = 0 \\ S^\dagger & \text{if } k\pi + (l-j)\frac{\pi}{2} \mod 2\pi = \frac{\pi}{2} \\ Z & \text{if } k\pi + (l-j)\frac{\pi}{2} \mod 2\pi = \pi \\ ZS^\dagger & \text{if } k\pi + (l-j)\frac{\pi}{2} \mod 2\pi = \frac{3\pi}{2} \end{cases} \tag{5.53}$$

This completes the proof of the lemma. □

5.5. Solving T-LMQC-EQUIV using the reduced Clifford set

In this section we describe an algorithm to check T-LMQC equivalence of graph states based on theorem 5.1. The results and notation are used from section 5.2 and 5.3, when one understands the methods described there, implementing the algorithm is not too hard. Either way, we will start in section 5.5.1 with a rough sketch before providing and discussing the algorithm in pseudo code. In section 5.5.2 we prove that the algorithm is correct. The runtime of the algorithm is discussed in section 5.5.3. An implementation in SAGE can be found on Github [12]. The main result of this section is given in theorem 5.2.

Theorem 5.2. *Algorithm 5 returns true if and only if the two input graph states are T-LMQC equivalent, and it returns false otherwise. The maximum number of qubits per node is restricted to 2. The running time of the algorithm is $\mathcal{O}\left(40^m|G|^4\right)$ where m is the number of two qubit nodes in T.*

5.5.1. The algorithm

Here we describe the T-LMQC equivalence algorithm based on theorem 5.1. First we discuss algorithm 4 which returns the graph state after gate teleportation for a given element of the reduced Clifford set. This is based on the calculations in appendix A.3. We will discuss the algorithm and the complexity line by line, however the intuition behind this algorithm follows from the calculations done in appendix A.3.

- Line 2: G and A_{GS}^U are connected by two edges, (a, A_0) and (b, A_1). This corresponds to the CZ gates in the gate teleportation circuit. We assume edge flips to be $\mathcal{O}(1)$ operations.
- Line 3: Pivot operations on (a, A_0) and (b, A_1), which correspond to the Hadamards from the Bell measurements.
- Line 4-7: If one of the corrections was a S gate, apply a local complementation on the corresponding ancilla qubit. Local complementations are $\mathcal{O}\left(N^2\right)$.
- Line 8: Remove the qubits a, b, A_0, A_1 from the graph as they are measured. Removing a vertex is an $\mathcal{O}(N)$ operation, as there are possible $N-1$ edges to delete as well.

Next, we will discuss the algorithm based on the reduced Clifford set step-by-step. The idea of the algorithm is to first (step 1) consider every multi-qubit node by

Algorithm 4 Modified gate teleportation

```
Input:  A graph $G = (V, E)$, two distinct vertices $a, b \in V[G]$,
        $A_{GS}^U \in \mathbf{L}_{T_A}(|Bell_2\rangle)$,$C_0, C_1 \in \{\mathbb{1}, S\}$.
Output: $|\tilde{G}\rangle$
function G.T.$^*$$(G, a, b, A_{GS}^U, C_0, C_1)$
    $|G_{tmp}\rangle \leftarrow |G + A_{GS}^U + (a, A_0) + (b, A_1)\rangle$
    $|G_{tmp}\rangle \leftarrow |\rho_{aA_0}\rho_{bA_1}(G_{tmp})\rangle$
    if $C_1 = S$ then
        $|G_{tmp}\rangle \leftarrow |\tau_{A_1}(G_{tmp})\rangle$
    if $C_0 = S^\dagger$ then
        $|G_{tmp}\rangle \leftarrow |\tau_{A_0}(G_{tmp})\rangle$
    $|G_{tmp}\rangle \leftarrow |G_{tmp} \backslash \{a, b, A_0, A_1\}\rangle$
    return $|G_{tmp}\rangle$
```

applying every local Clifford operation to $|G\rangle$, before checking SQC for the resulting state to $|G'\rangle$ (step 2). However, instead of checking all 11520 two qubit Cliffords, we will only consider the 40 elements from the reduced Clifford set defined in section 5.2. The two steps are discussed in more detail here.

1. In this step we will loop over all multi-qubit nodes in T, lets say there are m multi-qubit nodes. As different multi-qubit nodes are independent, i.e. one has to check all elements of the reduced Clifford set independent of the other node, this step can be visualized by a tree structure. The root is the graph $|G\rangle$ and by considering all elements of the reduced Clifford set for the first multi-qubit node in T the root has 40 leaves. When considering the next multi-qubit node in T, we only consider the leaves of the previous node and create 40 new leaves for every old leave. I.e., the branching factor is 40. After considering the last multi-qubit node in T, the tree has 40^m leaves and every leave is a graph state. The function in algorithm 4 is called to find the state after the gate teleportation.

2. After considering all multi-qubit nodes, we have a list $L^G_{\text{to Check}}$ (all leaves of the tree of step 1) of 40^m graph states. The final step is to one-by-one check the graphs in $L^G_{\text{to Check}}$ for SQC equivalence to $|G'\rangle$. If a success case is found, i.e. a graph state in $L^G_{\text{to Check}}$ is SQC-equivalent to $|G'\rangle$, we stop considering the remaining elements of $L^G_{\text{to Check}}$ and return true. Otherwise, if no element of $L^G_{\text{to Check}}$ is SQC-equivalent to $|G'\rangle$, the algorithm returns false.

The algorithm can be optimized, for example by traversing a tree and stopping if we have success,instead of first finding all elements of $L^G_{\text{to Check}}$. Another idea would be to keep a set of graphs instead of a list such that no duplicates are considered. However, this will not improve the worst case running time.

Algorithm 5 Is $|G\rangle$ T-LMQC equivalent to $|G'\rangle$

```
Input:  Graph states $|G\rangle$,          $|G'\rangle$ with $V[G] = V[G']$ and a partition $T$ of $V[G]$.
        Where $|i| \leq 2$ for $i \in T$.
Output: TRUE                    if $|G\rangle \sim_{T\text{-LMQC}} |G'\rangle$
        FALSE                   otherwise
if $|G\rangle \sim_{SQC} |G'\rangle$ then
    return TRUE
Set $L^G_{\text{to Check}} = list(|G\rangle)$
$multiQubitNodes$ = [$node$ for $node$ in $T$ if $|node| > 1$]
for $a, b \in multiQubitNodes$ do
    Set $tmplist = \emptyset$
    for $|G_{in}\rangle \in L^G_{\text{to Check}}$ do
        if $(a, b) \in |G_{in}\rangle$ then
            remove $(a, b)$ from $|G_{in}\rangle$    ▷Follows from assumption that $(a, b) \notin G$
        for $(A^U_{GS}, C_0, C_1) \in \mathcal{C}_2'^5$ do
            $|G_{\text{G.T.}}\rangle \leftarrow \text{G.T.}^*(G_{in}, A^U_{GS}, C_0, C_1)$
            append $|G_{\text{G.T.}}\rangle$ to $tmplist$
    Set $L^G_{\text{to Check}} \leftarrow tmplist$
Set $equiv$ = FALSE
for $|G_{in}\rangle \in L^G_{\text{to Check}}$ do
    if $|G_{in}\rangle \sim_{\text{SQC}} |G'\rangle$ then
        $equiv$ = TRUE
        break
return equiv
```

5.5.2. Proof that the algorithm is correct

In this section we prove that the algorithm presented in the previous section, algorithm 5, is correct. I.e., it returns true if and only if the two graph states are T-LMQC equivalent and false otherwise. Note that this proof will rely heavily on the definition of the reduced Clifford set, (definition 5.1) and theorem 5.1.

The algorithm succeeds

In this section we prove that algorithm 5 only returns true if and only if $|G\rangle \sim_{T-\text{LMQC}} |G'\rangle$. If the algorithm returns true, there is an element of the reduced Clifford set which results in a $|G_{in}\rangle$ SQC-equivalent to $|G'\rangle$. $|G_{in}\rangle$ is achieved by gate teleporting, which is essentially the same as applying the gate directly to $|G\rangle$. Thus, when the algorithm returns true, $|G\rangle$ can be transformed to $|G'\rangle$ by only T-LMQC operations. Therefore, if the algorithm returns TRUE, it holds that $|G\rangle \sim_{T-\text{LMQC}} |G'\rangle$.

The algorithms fails

In this section we prove that algorithm 5 only returns false if and only if $|G\rangle \nsim_{T-\text{LC}} |G'\rangle$. By definition the reduced Clifford set covers all multi-qubit Cliffords up to SQC equivalence. Therefore, if the algorithm returns false, there exists no local multi-qubit Clifford taking $|G\rangle$ to $|G'\rangle$.

5.5.3. Runtime of the algorithm

Algorithm 5 decides T-LMQC equivalence of graph states. Here we show that the runtime is $\mathcal{O}(40^m N^4)$ where m is the number of two qubit nodes in T and $N = |V[G]|$. Therefore, algorithm 5 is first analysed line by line. Afterwards, the computational heavy steps can be bundled to find the runtime.

- Line 1-2: Checking SQC-equivalence can be done in $\mathcal{O}(N^4)$ according to theorem 3.2.
- Line 3,6,12,13,14,17: These are all operations which can be done in constant time, $\mathcal{O}(1)$.
- Line 4: In the worst case this loop runs over N elements, so $\mathcal{O}(N)$.
- Line 5-13: The outer for loop runs over m elements. Let us use i to denote the ith iteration of the outer loop (first iteration is $i = 0$). Then the for loop starting in line 7 runs over 40^i graphs, as this is the number of leaves of the previous outer loop (remember the tree form discussed in step 1 of the algorithm). Line 8-9 is be done in constant time. The inner-most for

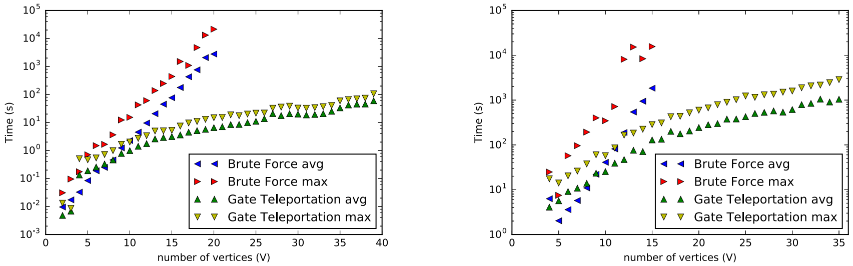
loop always considers all 40 elements of the reduced Clifford set. Line 11 consists of local complementations ($\mathcal{O}(N^2)$ [11]), edge flips ($\mathcal{O}(1)$) and vertex deletions ($\mathcal{O}(N)$). Therefore, the total complexity of the inner-most for loop is $\mathcal{O}(40N^2)$. The for loop starting in line 7 has time complexity $\mathcal{O}(40^i 40N^2)$. This results in the total complexity of the outer for loop to be $\mathcal{O}(40^m N^2)$.

- Line 15-18: The list $L^G_{\text{to Check}}$ has 40^m elements. Checking SQC-equivalence can be done in $\mathcal{O}(N^4)$, and this has to be checked for every element of $L^G_{\text{to Check}}$. Therefore the total running time is $\mathcal{O}(40^m N^4)$.

Thus, the total running time of algorithm 5 is $\mathcal{O}(40^m N^4)$. In section 5.5.4 the performance of the algorithm is compared to algorithm 2 (brute-force) for an actual implementation of the two algorithms in SAGE.

5.5.4. Benchmarking

In this section the performance of algorithm 2 and algorithm 5 are compared by considering running times of implementations of both algorithms in SAGE. Note that for figure 5.13a (b), the number of multi-qubit nodes is fixed to 1 (2) . The x-axis denotes an increasing number of vertices (single qubit nodes). The ”brute-force” algorithm clearly scales exponentially, whereas the gate teleportation algorithm grows significantly slower. This is as expected after the running time analysis of both algorithms.



(a) One two-qubit node and single qubit nodes

(b) Two two-qubit nodes and single qubit nodes

Figure 5.13: Actual running times of two algorithms for the T-LMQC-EQUIV problem with a) $T = (\{0,1\},\dots,\{V-1\})$ and b) $T = (\{0,1\},\{2,3\},\dots,\{V-1\})$. ”Brute force” is algorithm 2, where ”Gate teleportation” is algorithm 5. The max and average running times are calculated from at least 50 instances of the T-LMQC problem. The input graph states are random connected graphs, the partition is such that there is a) one or b) two two-qubit node(s) and all other nodes are single qubit nodes.

5.6. Extension to general multi-qubit nodes

The concept behind the reduced Clifford set in section 5.2 and the corresponding algorithm 5 are not restricted the bound of 2 qubits per node. However, due to the exponential scaling in the number of qubits N of 1) the N-qubit Clifford group and 2) the set of all graphs on N vertices, finding the correct set of graphs and operations of the reduced Clifford set becomes very involved.

The operations are still doable, it is the same concept we used to find the operations needed on 2 qubits, although there are more qubits involved. In the reduced Clifford set of theorem 5.1, we know that $C_i \in \{\mathbb{1}, S\}$, for $i \in \{0, 1\}$. One might think that this is also true for $i \in \{0, 1, 2\}$ in the case of 3 or more qubits per node. However, the calculations behind the result in theorem 5.1 are very subtle. It might be that there is a step in the proof that has to be reconsidered in order to cover the Clifford group on 3 qubits.

In order to find the reduced Clifford set on 3 qubits, we have to consider the 3-Bell pairs state shown in figure 5.14. Finding the set of all graphs in $\mathbf{L}_{Bell_3}(T)$ is however quite difficult. Note that there are $2^{\frac{6\times 5}{2}} = 32768$ possible graphs on 6 qubits and that $|\mathcal{C}_3| = 92897280$. However, by making use of the computer facilities of the INSY group at EWI, it was possible to find the number of graphs T-LMQC equivalent to the 3-Bell pair state. The results are shown in table 5.1.

M	$\left\lvert\mathcal{C}'_M\right\rvert$
2	24
3	10760

Table 5.1: The number of graph states T-LMQC equivalent to M Bell pairs all contributing one qubit to a M-qubit node.

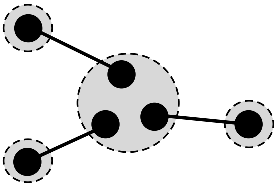


Figure 5.14: The M-Bell pairs state for $M = 3$, where every Bell pair contributes one qubit to multi-qubit node.

References

[1] S. Wehner, D. Elkouss, and R. Hanson, *Quantum internet: A vision for the road ahead,* Science **362** (2018).

[2] A. Ekert and R. Renner, *The ultimate physical limits of privacy,* Nature **507**, 443 (2014).

[3] A. Broadbent, J. Fitzsimons, and E. Kashefi, *Universal blind quantum computation,* in *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on* (IEEE, 2009) pp. 517–526.

[4] D. Gottesman, T. Jennewein, and S. Croke, *Longer-baseline telescopes using quantum repeaters,* Physical review letters **109**, 070503 (2012).

[5] P. W. Shor, *Algorithms for quantum computation: Discrete logarithms and factoring,* in *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on* (Ieee, 1994) pp. 124–134.

[6] D. Markham and B. C. Sanders, *Graph states for quantum secret sharing,* Physical Review A **78**, 042309 (2008).

[7] R. Raussendorf and H. J. Briegel, *A one-way quantum computer,* Physical Review Letters **86**, 5188 (2001).

[8] M. Van den Nest, J. Dehaene, and B. De Moor, *Graphical description of the action of local clifford transformations on graph states,* Physical Review A **69**, 022316 (2004).

[9] Z. Ji, J. Chen, Z. Wei, and M. Ying, *The lu-lc conjecture is false,* arXiv preprint arXiv:0709.1266 (2007).

[10] A. Dahlberg, J. Helsen, and S. Wehner, *How to transform graph states using single-qubit operations: computational complexity and algorithms,* arXiv preprint arXiv:1805.05306 (2018).

[11] A. Bouchet, *An efficient algorithm to recognize locally equivalent graphs,* Combinatorica **11**, 315 (1991).

[12] *Git-repository with implemented code* , https://github.com/Cmolengraaf/vertex-minors.

[13] H. P. Nautrup, N. Friis, and H. J. Briegel, *Fault-tolerant interface between quantum memories and quantum processors,* Nature Communications **8**, 1321 (2017).

[14] T. Jochym-O'Connor, A. Kubica, and T. J. Yoder, *Disjointness of stabilizer codes and limitations on fault-tolerant logical gates,* Physical Review X **8**, 021047 (2018).

[15] M. A. Nielsen and I. Chuang, *Quantum computation and quantum information,* (2002).

[16] D. Gottesman, *Stabilizer codes and quantum error correction," 1997, Caltech Ph. D*, Ph.D. thesis, dissertation, eprint: quant-ph/9705052.

[17] M. Ozols, *Clifford group,* (2008).

[18] M. Hein, W. Dür, J. Eisert, R. Raussendorf, M. Nest, and H.-J. Briegel, *Entanglement in graph states and its applications,* arXiv preprint quant-ph/0602096 (2006).

[19] J. Dehaene and B. De Moor, *The Clifford group, stabilizer states, and linear and quadratic operations over GF(2),* Physical Review A - Atomic, Molecular, and Optical Physics **68**, 10 (2003).

[20] *SimulaQron,* https://www.simulaqron.org/.

[21] D. Gottesman and I. L. Chuang, *Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations,* Nature **402**, 390 (1999).

[22] L. E. Danielsen, *On self-dual quantum codes, graphs, and boolean functions,* arXiv preprint quant-ph/0503236 (2005).

[23] Y.-C. Zheng, C.-Y. Lai, T. A. Brun, and L.-C. Kwek, *Depth reduction for quantum Clifford circuits through Pauli measurements,* (2018).

[24] A. Cayley, *A theorem on trees,* Quart. J. Math. **23**, 376 (1889).

A

Appendix

A.1. Examples of SAGE implementation

Here we discuss some examples of how to use our SAGE implementation for testing T-LMQC equivalence, which is available on our repository [12].

The file LMQC.sage contain classes for sage that can be used to study graph states under local multi-qubit Clifford operations. Using the class SimpleGraphLMQC one can test whether two graphs states are T-LMQC equivalent. To use the classes, start sage and load the script by writing `load(”LMQC.sage”)`. Some useful methods are described below:

- `SimpleGraphLMQC(data)`: Creates an instance of the class `SimpleGraphLMQC`. `data` can be a an instance of the `Graph`-class already in SAGE, a dictionary describing the neighbors of vertices, etc.
- `G.is_LC_eq(Gp)`: This checks if G and Gp are SQC equivalent. Note that LC does not refer to local multi-qubit Cliffords here.
- `G.set_partition(T)`: Initializes the partition of `G` as `T`. This is needed for `G.is_LMQC_eq(Gp)`.
- `G.is_LMQC_eq(Gp,method=”brute”)`: Checks if `G` is T-LMQC equivalent to `Gp`. Note that there are two other methods with an improved runtime, `gate_tele` and `conj`. These three methods correspond to the algorithms described in this thesis. So:

  `method=”brute”` → Algorithm 2

method=”conj”’ → Algorithm 3 (number of qubits per node ≤ 2)
method=”gate_tele” → Algorithm 5 (number of qubits per node ≤ 2)

Example 1

In this example G is the complete graph on 4 vertices and Gp is the star graph with center 0 from which the edge $(0,1)$ is removed. Note that the star graph and complete graph are SQC equivalent, and that vertices 0 and 1 are inside the same node. Therefore, we expect G and Gp to be T-LMQC equivalent.

```
G=SimpleGraphLMQC([0,1,2,3])
G.set_partition([[0,1],[2],[3]])
Gp = SimpleGraph({0:[2,3],1:[]})
G.is_LMQC_eq(Gp)
```

Example 2

In this example T-LMQC equivalence is decided using all three algorithms discussed in this thesis, for two randomly generated graphs. To generate a random graph, the function `graphs.RandomGNP(V,p)` is used. Here `V` stands for the number of vertices, and `p` is the probability that an edge is inserted in the graph. Note that `G.is_LMQC_eq(Gp,method=’conj’)` will return an error message if `G` or `Gp` is not a locally connected graph.

```
G = SimpleGraphLMQC(graphs.RandomGNP(4,0.7))
Gp = SimpleGraph(graphs.RandomGNP(4,0.9))
G.set_partition([[0],[1],[2,3]])
G.is_LMQC_eq(Gp,method=’brute’)
G.is_LMQC_eq(Gp,method=’gate_tele’)
G.is_LMQC_eq(Gp,method=’conj’)
```

Example 3

In this example T-LMQC equivalence is decided for a graph state on 6 qubits with one three qubit node. Note there is only one algorithm able to solve this, and that it might take a long time. Note that this could take a long time before returning a result.

```
G = SimpleGraphLMQC(Graph({0:[3],1:[4],2:[5]}))
Gp = SimpleGraph(graphs.RandomGNP(6,0.9))
G.set_partition([[0,1,2],[3],[4],[5]])
G.is_LMQC_eq(Gp,method=’brute’)
```

A.2. Generating random connected graphs

In this section we describe how to generate a random connected graph G on N vertices. There are three main steps in doing this.

1. Generate a random tree G on N vertices.
2. Choose the total number of edges, $E_{\max}$, G should have from the set $\{N - 1, \cdots, N(N-1)/2\}$
3. Add random edges to G till $\left|E[G]\right| = E_{\max}$.

We will now explain each step in a bit more detail. It is shown [24] that there are N^{N-2} trees on the vertex set $\{0, \cdots, N-1\}$. Choosing one of these trees uniformly at random corresponds to generating a random tree. Note that in SAGE there is method for a $graphs$ object which returns a random tree ($graphs.randomTree(N)$). The next step is to choose the total number of edges uniformly at random from the set $\{N-1, \cdots, N(N-1)/2\}$. Note that there exists no simple connected graph of which the number of vertices is outside this set. In the final step, we start by making a set E_c of all edges not present in G. While $\left|E[G]\right| \leq E_{\max}$, we choose an edge (a, b) from E_c, remove this edge from E_c and add it to G.

A.3. Stabilizer state after gate teleportation calculations

In this section the calculations are performed for finding the stabilizer state after the gate teleportation circuit, which correspond to the state in equation 5.13. Note that we assume that $(a, b) \notin G$.

A.3.1. $C_0 = \mathbb{1}, C_1 = \mathbb{1}$

- In this section we will calculate the state after the gate teleportation circuit for the case that $C_0, C_1 = \mathbb{1}$. The first step is to write down the state from the definition in equation 5.13.

$$\text{G.T.}^*\left(G, A_G^{U'}, \mathbb{1}, \mathbb{1}\right) = \left\langle B_1^{bA_1}\right| \left\langle B_1^{aA_0}\right| \mathbb{1}_2 \otimes \mathbb{1}_2 \left|G + A_G^{U'}\right\rangle$$

- As in this case the operations act trivially, the next step is to expand the Bell measurements in order to see how this operations affect the graph state.

$$= \left\langle 0_{A_0} 0_{A_1} 0_a 0_b \right| H_{A_0} H_{A_1} H_a H_b CZ_{aA_0} CZ_{bA_1} \left|G + A_G^{U'}\right\rangle$$

- As G and A_G^U are by definition not connected, the edges (a, A_0), (b, A_1) are added. As $(a,b) \notin G$, the four Hadamards always act as two pivots following 5.4. However, from lemma 3.1 we know that there are also possible Z operations when writing 2 Hadamards as a pivot. Using lemma 5.3 it is known that any Pauli before measurements, only result in Pauli's afterwards. Therefore, we have the following state up to Pauli's:

$$= \left\langle 0_{A_0} 0_{A_1} 0_a 0_b \middle| \rho_{aA_0} \rho_{bA_1} (G + A_G^{U'} + (a, A_0) + (b, A_1)) \right\rangle$$

- The last step is the measurements of the four qubits, which removes them from the graph state.

$$= \left| \rho_{aA_0} \rho_{bA_1} (G + A_G^{U'} + (a, A_0) + (b, A_1)) \backslash \{A_0, A_1, a, b\} \right\rangle$$

A.3.2. $C_0 = H, C_1 = \mathbb{1}$

- For $C_0 = H$, $C_1 = \mathbb{1}$ we start from the definition.

$$\text{G.T.}^* \left(G, A_G^{U'}, H, \mathbb{1} \right) = \left\langle B_1^{bA_1} \middle| \left\langle B_1^{aA_0} \middle| H_{A_0} \middle| G + A_G^{U'} \right\rangle \right.$$

- As we know how 2 Hadamards act on a graph state, we add a Hadamard on a not-measured neighbour of A_0, in this case u. I.e., $u \in N_{A_0}(A_G^{U'}) \backslash A_1$

$$= \left\langle B_1^{bA_1} \middle| \left\langle B_1^{aA_0} \middle| H_u H_u H_{A_0} \middle| G + A_G^{U'} \right\rangle \right.$$

- As H_u commutes with the measurements of other qubits, we have:

$$= H_u \left\langle B_1^{bA_1} \middle| \left\langle B_1^{aA_0} \middle| G + \rho_{A_0 u}(A_G^{U'}) \right\rangle \right.$$

- Here one can recognize the stabilizer state after measurements for $C_0, C_1 = \mathbb{1}$ for ancilla state A_G^U, up to H_u.

$$= H_u \text{G.T.}^* \left(G, \rho_{A_0 u}(A_G^{U'}), \mathbb{1}, \mathbb{1} \right)$$

A.3.3. $C_0 = \mathbb{1}, C_1 = H$

See section A.3.2 for details on these calculations.

$$\begin{aligned}
&\text{G.T.}^*\left(G, A_G^{U'}, \mathbb{1}, H\right)\\
=&\left\langle B_1^{bA_1}\right|\left\langle B_1^{aA_0}\right| H_{A_1}\left|G + A_G^{U'}\right\rangle\\
=&\left\langle B_1^{bA_1}\right|\left\langle B_1^{aA_0}\right| H_u H_u H_{A_1}\left|G + A_G^{U'}\right\rangle\\
=&H_u\left\langle B_1^{bA_1}\right|\left\langle B_1^{aA_0}\right|G + \rho_{A_1 u}(A_G^{U'})\right\rangle\\
=&H_u\text{G.T.}^*\left(G, \rho_{A_1 u}(A_G^{U'}), \mathbb{1}, \mathbb{1}\right)
\end{aligned}$$

$$u \in N_{A_1}(A_G^{U'}) \backslash A_1$$

A.3.4. $C_0 = S, C_1 = \mathbb{1}$

- Again we start from the definition:

$$\text{G.T.}^*\left(G, A_G^{U'}, S, \mathbb{1}\right) = \left\langle B_1^{bA_1}\right|\left\langle B_1^{aA_0}\right| S_{A_0}^\dagger\left|G + A_G^{U'}\right\rangle$$

- The question is now how $S_{A_0}^\dagger$ affects the graph state. As it is not clear how $S_{A_0}^\dagger$ changes the graph state directly, we commute it through the Bell measurement operations. Note that S and CZ commute, and that $HS^\dagger = \sqrt{X}^\dagger H$. Then we find:

$$=\langle 0_{A_0} 0_{A_1} 0_a 0_b| \sqrt{X_{A_0}}^\dagger H_{A_0} H_{A_1} H_a H_b CZ_{bA_1} CZ_{aA_0}\left|G + A_G^{U'}\right\rangle$$

- Using $\sqrt{X}\,|0\rangle = |{+i}\rangle$ we change the measurement basis of qubit A_0, however the other three qubits are still measured in the Z-basis. Therefore, we first do the measurements of a, b and A_1, to find:

$$=\left\langle +i_{A_0}\middle|\rho_{aA_0}\rho_{bA_1}(G + A_G^{U'} + (a, A_0) + (b, A_1))\backslash\{A_1, a, b\}\right\rangle$$

The last step is to measure qubit A_0 and find the resulting graph state using the measurement rules in definiton 3.4.

$$=\sqrt{-iZ}^{N_{A_0}}\left|\tau_{A_0}\rho_{aA_0}\rho_{bA_1}(G + A_G^{U'} + (a, A_0) + (b, A_1))\backslash\{A_0, A_1, a, b\}\right\rangle$$

A.3.5. $C_0 = \mathbb{1}, C_1 = S$

See section A.3.4 for the idea behind these calculations.

$$
\begin{aligned}
&\text{G.T.}^*\left(G, A_G^{U'}, \mathbb{1}, S\right)\\
&=\left\langle B_1^{bA_1}\right|\left\langle B_1^{aA_0}\right| S_1^\dagger \left|G + A_G^{U'}\right\rangle\\
&=\langle 0_{A_0} 0_{A_1} 0_a 0_b |\, H_b H_{A_1} H_a H_{A_0} CZ_{bA_1} CZ_{aA_0} S_1^\dagger \left|G + A_G^{U'}\right\rangle\\
&=\langle 0_{A_0} 0_{A_1} 0_a 0_b |\, H_{A_0} S_1^\dagger H_{A_1} H_a H_b CZ_{bA_1} CZ_{aA_0} \left|G + A_G^{U'}\right\rangle\\
&=\langle 0_{A_0} 0_{A_1} 0_a 0_b |\, \sqrt{X_{A_1}}^\dagger\, H_{A_0} H_{A_1} H_a H_b CZ_{bA_1} CZ_{aA_0} \left|G + A_G^{U'}\right\rangle\\
&=\left\langle 0_{A_0} + i_{A_1} 0_a 0_b \middle| \rho_{aA_0} \rho_{bA_1} (G + A_G^{U'} + (a, A_0) + (b, A_1))\right\rangle\\
&=\left\langle +i_{A_1} \middle| \rho_{aA_0} \rho_{bA_1} (G + A_G^{U'} + (a, A_0) + (b, A_1)) \setminus \{A_0, a, b\}\right\rangle\\
&=\sqrt{-iZ}^{N_{A_1}} \left|\tau_{A_1} \rho_{aA_0} \rho_{bA_1} (G + A_G^{U'} + (a, A_0) + (b, A_1)) \setminus \{A_0, A_1, a, b\}\right\rangle
\end{aligned}
$$

A.3.6. $C_0 = HS, C_1 = \mathbb{1}$

- Equation 5.13 is our starting point.

$$
\text{G.T.}^*\left(G, A_G^{U'}, HS, \mathbb{1}\right) = \left\langle B_1^{bA_1}\right|\left\langle B_1^{aA_0}\right| H_{A_0} S_{A_0}^\dagger \left|G + A_G^{U'}\right\rangle
$$

As it is not directly clear how $H_{A_0} S_{A_0}^\dagger$ act on a graph state, we will rewrite. First we do an extra operation $S_{A_0} S_{A_0}^\dagger = \mathbb{1}$ just before the CZ gates. S_{A_0} we then pull all the way to the left to the projectors. $S_{A_0}^\dagger$ is used to get an operation for which we might know the effect on a graph state: $\sqrt{-iX_{A_0}} = S_{A_0}^\dagger H_{A_0} S_{A_0}^\dagger$. Now remember from section 3.2 that $\sqrt{-iX_j}$ is the operation on qubit j corresponding to a local complementation on j. Therefore, we find the following:

$$
= \langle 0_{A_0} 0_{A_1} 0_a 0_b |\, \sqrt{X_{A_0}} H_{A_0} H_{A_1} H_a H_b CZ_{aA_0} CZ_{bA_1} \sqrt{-iX_{A_0}} \left|G + A_G^{U'}\right\rangle
$$

In the next step we use definition 3.1, the definition of how local complementing acts on a graph state, to find:

$$
= \langle 0_{A_0} 0_{A_1} 0_a 0_b |\, \sqrt{X_{A_0}} H_{A_0} H_{A_1} H_a H_b CZ_{aA_0} CZ_{bA_1} \sqrt{-iZ}^{N_{A_0}(A_G^{U'})} \left|G + \tau_{A_0}(A_G^{U'})\right\rangle
$$

Doing a local complementation includes SQC operations on the neighbourhood of the vertex, so in this case on N_{A_0}. The only measured qubit which might be in N_{A_0} is A_1, therefore the measurement basis of qubit A_1 is changed if $(A_0, A_1) \in A_G^U$.

$$=\sqrt{-iZ}^{N_{A_0}(A_G^{U'})\backslash A_1}$$

$$\langle 0_{A_0} 0_{A_1} 0_a 0_b | \sqrt{X_{A_0}} \sqrt{-iX_{A_1}}^{[(A_0,A_1)\in A_G^{U'}]} H_{A_0} H_{A_1} H_a H_b CZ_{aA_0} CZ_{bA_1} \left| G + \tau_{A_0}(A_G^{U'}) \right\rangle$$

As the measurements of a, b are the easiest, we will do them first. Afterwards, we use the 2 qubit measurement rules from section 5.4.2 to find the corresponding graph states (up to SQC operations on the final graph state)

$$\sim_{SQC} \begin{cases} \left| \tau_{A_0} \tau_{A_1} \rho_{aA_0} \rho_{bA_1} \left(G + \tau_{A_0}(A_G^{U'}) + (a, A_0) + (b, A_1) \right) \backslash \{a, b, A_0, A_1\} \right\rangle \\ \text{if } (A_0, A_1) \in A_G^{U'} \\ \\ \left| \tau_{A_0} \rho_{aA_0} \rho_{bA_1} \left(G + \tau_{A_0}(A_G^{U'}) + (a, A_0) + (b, A_1) \right) \backslash \{a, b, A_0, A_1\} \right\rangle \\ \text{if } (A_0, A_1) \notin A_G^{U'} \end{cases}$$

A.3.7. $C_0 = \mathbb{1}, C_1 = HS$

This is similar as section A.3.6 when swapping C_0 and C_1.

$$\text{G.T.}^* \left(G, A_G^{U'}, \mathbb{1}, HS \right) = \left\langle B_1^{bA_1} \right| \left\langle B_1^{aA_0} \right| H_{A_1} S_{A_1}^\dagger \left| G + A_G^{U'} \right\rangle$$

$$\sim_{SQC} \begin{cases} \text{if } (A_0, A_1) \in A_G^{U'} & \left| \tau_{A_0} \tau_{A_1} \rho_{aA_0} \rho_{bA_1} \left(G + \tau_{A_1}(A_G^{U'}) + (a, A_0) + (b, A_1) \right) \backslash \{A_0, A_1, a, b\} \right\rangle \\ \text{if } (A_0, A_1) \notin A_G^{U'} & \left| \tau_{A_1} \rho_{aA_0} \rho_{bA_1} \left(G + \tau_{A_1}(A_G^{U'}) + (a, A_0) + (b, A_1) \right) \backslash \{A_0, A_1, a, b\} \right\rangle \end{cases}$$

A.3.8. $C_0 = H, C_1 = H$

- As before, we start from the definition.

$$\text{G.T.}^* \left(G, A_G^{U'}, H, H \right) = \left\langle B_1^{bA_1} \right| \left\langle B_1^{aA_0} \right| H_{A_0} H_{A_1} \left| G + A_G^{U'} \right\rangle$$

- It is known how two Hadamards act on a graph state if the corresponding edge is present in the graph, so in this case $(A_0, A_1) \in A_G^U$. That is the first case we distinguish. The second and thirth case are a little bit more tricky as it is not known how two Hadamards on non-adjacent vertices transform a graph state. However, it turns out that when always starting with a pivot on the vertex with only 1 neighbour, the vertex still to pivot is always connected to A_a or A_b or both. This leads to:

$$= \begin{cases} \text{if } (A_0, A_1) \in A_G^U & Z[N_{A_0} \cap N_{A_1}] \left\langle B_1^{bA_1} \middle| \left\langle B_1^{aA_0} \middle| \middle| G + \rho_{A_0 A_1}(A_G^{U'}) \right\rangle\right.\right. \\ \quad \text{elif } |N_{A_0}| = 1 & \left\langle B_1^{bA_1} \middle| \left\langle B_1^{aA_0} \middle| H_u H_{A_1} \middle| G + \rho_{A_0 u}(A_G^{U'}) \right\rangle\right.\right. \\ \quad \text{elif } |N_{A_1}| = 1 & \left\langle B_1^{bA_1} \middle| \left\langle B_1^{aA_0} \middle| H_u H_{A_0} \middle| G + \rho_{A_1 u}(A_G^{U'}) \right\rangle\right.\right. \end{cases}$$

- if $|N_{A_0}| = 1$ then $u \in N_{A_0}(A_G^{U'})$, if $|N_{A_1}| = 1$ then $u \in N_{A_1}(A_G^{U'})$. And by choosing v similairly, so if $|N_{A_0}| = 1$ then $v \in N_{A_1}(\rho_{A_0 u}(A_G^{U'}))$, if $|N_{A_1}| = 1$ then $v \in N_{A_0}(\rho_{A_1 u}(A_G^{U'}))$:

$$= \begin{cases} \text{if } (A_0, A_1) \in A_G^U & Z[N_{A_0} \cap N_{A_1}] \left\langle B_1^{bA_1} \middle| \left\langle B_1^{aA_0} \middle| \middle| G + \rho_{A_0 A_1}(A_G^{U'}) \right\rangle\right.\right. \\ \quad \text{elif } |N_{A_0}| = 1 & \left\langle B_1^{bA_1} \middle| \left\langle B_1^{aA_0} \middle| H_v H_u \middle| G + \rho_{A_1 v}\rho_{A_0 u}(A_G^{U'}) \right\rangle\right.\right. \\ \quad \text{elif } |N_{A_1}| = 1 & \left\langle B_1^{bA_1} \middle| \left\langle B_1^{aA_0} \middle| H_v H_u \middle| G + \rho_{A_0 v}\rho_{A_1 u}(A_G^{U'}) \right\rangle\right.\right. \end{cases}$$

As u, v are chosen such that they always commute with the projectors, all three cases are equal to the state in section A.3.1 up to SQC.

$$\sim_{SQC} \text{G.T.}^*(G, \tilde{A}_G^{U'}, \mathbb{1}, \mathbb{1})$$

A.3.9. $C_0 = H, C_1 = S$

This case builds further on the results of section A.3.4 and A.3.2.

$$\begin{aligned} \text{G.T.}^*\left(G, A_G^{U'}, H, S\right) &= \left\langle B_1^{bA_1} \middle| \left\langle B_1^{aA_0} \middle| H_{A_0} S_1^\dagger \middle| G + A_G^{U'} \right\rangle\right.\right. \\ &= H_u \left\langle B_1^{bA_1} \middle| \left\langle B_1^{aA_0} \middle| S_1^\dagger \middle| G + \rho_{A_0 u}(A_G^{U'}) \right\rangle\right.\right. \\ &= H_u \text{G.T.}^*(G, \rho_{A_0 u}(A_G^{U'}), \mathbb{1}, S) \end{aligned}$$

$HS^\dagger = \sqrt{X}^\dagger H$

$$u \in N_{A_0}(A_G^{U'}) \backslash A_1$$

A.3.10. $C_0 = H, C_1 = HS$

This case builds further on the results of section A.3.4 and A.3.2.

$$\begin{aligned}
\text{G.T.}^* &\left(G, A_G^{U'}, H, HS\right) \\
&= \left\langle B_1^{bA_1} \right| \left\langle B_1^{aA_0} \right| H_{A_0} H_{A_1} S_1^\dagger \left| G + A_G^{U'} \right\rangle \\
&= H_u \left\langle B_1^{bA_1} \right| \left\langle B_1^{aA_0} \right| H_{A_1} S_1^\dagger \left| G + \rho_{A_0 u}(A_G^{U'}) \right\rangle \\
&= H_u \text{G.T.}^* \left(G, \rho_{A_0 u}(A_G^{U'}), \mathbb{1}, HS^\dagger\right)
\end{aligned}$$

$$u \in N_{A_0}(A_G^{U'}) \backslash A_1$$

A.3.11. $C_0 = S, C_1 = H$

This case builds further on the results of section A.3.4 and A.3.2.

$$\begin{aligned}
&\text{G.T.}^* \left(G, A_G^{U'}, S, H\right) \\
&= \left\langle B_1^{bA_1} \right| \left\langle B_1^{aA_0} \right| H_{A_1} S_{A_0}^\dagger \left| G + A_G^{U'} \right\rangle \\
&= H_u \left\langle B_1^{bA_1} \right| \left\langle B_1^{aA_0} \right| S_{A_0}^\dagger \left| G + \rho_{A_1 u}(A_G^{U'}) \right\rangle \\
&= H_u \text{G.T.}^* \left(G, \rho_{A_1 u}(A_G^{U'}), S^\dagger, \mathbb{1}\right)
\end{aligned}$$

$$u \in N_{A_1}(A_G^{U'}) \backslash A_0$$

A.3.12. $C_0 = HS, C_1 = H$

This case builds further on the results of section A.3.2 and A.3.6.

$$\begin{aligned}
&\text{G.T.}^* \left(G, A_G^{U'}, HS, H\right) \\
&= \left\langle B_1^{bA_1} \right| \left\langle B_1^{aA_0} \right| H_{A_0} S_{A_0}^\dagger H_{A_1} \left| G + A_G^{U'} \right\rangle \\
&= H_u \left\langle B_1^{bA_1} \right| \left\langle B_1^{aA_0} \right| H_{A_0} S_{A_0}^\dagger \left| G + \rho_{A_1 u}(A_G^{U'}) \right\rangle \\
&= H_u \text{G.T.}^* \left(G, \rho_{A_1 u}(A_G^{U'}), HS, \mathbb{1}\right)
\end{aligned}$$

$$u \in N_{A_0}(A_G^{U'}) \backslash A_1$$

A.3.13. $C_0 = S, C_1 = S$

- Here we will do something very similar to section A.3.4, but now for 2 qubits instead of only one. Thus we refer to section A.3.4 for an explanation of this calculations. However, the final step towards the last line requires a little bit more attention. There, the two qubit measurement rules of section 5.4.2 are used, together with lemma 5.5.

$$
\begin{aligned}
&\text{G.T.}^*\left(G, A_G^{U'}, S, S\right)\\
&=\left\langle B_1^{bA_1}\right|\left\langle B_1^{aA_0}\right| S_{A_0}^\dagger S_{A_1}^\dagger \left|G + A_G^{U'}\right\rangle\\
&=\langle 0_{A_0} 0_{A_1} 0_a 0_b|\, H_b H_{A_1} H_a H_{A_0} CZ_{bA_1} CZ_{aA_0} S_{A_0}^\dagger S_{A_1}^\dagger \left|G + A_G^{U'}\right\rangle\\
&=\langle 0_{A_0} 0_{A_1} 0_a 0_b|\, H_{A_0} S_{A_0}^\dagger H_{A_1} S_{A_1}^\dagger H_a H_b CZ_{bA_1} CZ_{aA_0} \left|G + A_G^{U'}\right\rangle\\
&=\langle 0_{A_0} 0_{A_1} 0_a 0_b|\, \sqrt{X_{A_0}}^\dagger \sqrt{X_{A_1}}^\dagger H_{A_0} H_{A_1} H_a H_b CZ_{bA_1} CZ_{aA_0} \left|G + A_G^{U'}\right\rangle\\
&=\left\langle +i_{A_0} + i_{A_1} 0_a 0_b \middle| \rho_{aA_0}\rho_{bA_1}\left(G + A_G^{U'} + (a, A_0) + (b, A_1)\right)\right\rangle\\
&=\left\langle +i_{A_0} + i_{A_1} \middle| \rho_{aA_0}\rho_{bA_1}\left(G + A_G^{U'} + (a, A_0) + (b, A_1)\right)\backslash a, b\right\rangle\\
\sim_{SQC}&\left|\tau_{A_0}\tau_{A_1}\rho_{aA_0}\rho_{bA_1}\left(G + A_G^{U'} + (a, A_0) + (b, A_1)\right)\backslash\{A_0, A_1, a, b\}\right\rangle
\end{aligned}
$$

A.3.14. $C_0 = S, C_1 = HS$

- We start from the definition:

$$
\text{G.T.}^*\left(G, A_G^{U'}, S, HS\right) = \left\langle B_1^{bA_1}\right|\left\langle B_1^{aA_0}\right| S_{A_0}^\dagger H_{A_1} S_{A_1}^\dagger \left|G + A_G^{U'}\right\rangle
$$

- The first steps are very similar to section A.3.6 and A.3.4, by following the procedure described there we find:

$$
\begin{aligned}
&=\sqrt{-iZ}^{N_{A_1}(A_G^{U'})\backslash A_0}\\
&\qquad \langle i_{A_0} 0_{A_1} 0_a 0_b|\, \sqrt{X_{A_1}}\sqrt{-iX_{A_0}}^{[(A_0, A_1) \in A_G^{U'}]} H_{A_0} H_{A_1} H_a H_b CZ_{aA_0} CZ_{bA_1} \left|G + \tau_{A_1}(A_G^{U'})\right\rangle
\end{aligned}
$$

- Writing the two cases seperatly and applying the projectors on a and b this leads to:

$$= \begin{cases} \sqrt{-iZ}^{N_{A_1}(A_G^{U'})\backslash A_0} \left\langle 0_{A_0} - i_{A_1} \right| \left| \rho_{aA_0}\rho_{bA_1}\left(G + \tau_{A_1}(A_G^{U'}) + (a, A_0) + (b, A_1)\right) \backslash a, b \right\rangle \\ \hfill \text{if } (A_0, A_1) \in A_G^{U'} \\ \\ \sqrt{-iZ}^{N_{A_1}(A_G^{U'})} \left\langle i_{A_0} - i_{A_1} \right| \left| \rho_{aA_0}\rho_{bA_1}\left(G + \tau_{A_1}(A_G^{U'}) + (a, A_0) + (b, A_1)\right) \backslash a, b \right\rangle \\ \hfill \text{if } (A_0, A_1) \notin A_G^{U'} \end{cases}$$

- Then using the two qubit measurement rules from section 5.4.2 and lemma 5.5 this leads to:

$$\sim_{SQC} \begin{cases} \left| \tau_{A_1}\rho_{aA_0}\rho_{bA_1}\left(G + \tau_{A_1}(A_G^{U'}) + (a, A_0) + (b, A_1)\right) \backslash \{A_0, A_1, a, b\} \right\rangle \\ \hfill \text{if } (A_0, A_1) \in A_G^{U'} \\ \\ \left| \tau_{A_0}\tau_{A_1}\rho_{aA_0}\rho_{bA_1}\left(G + \tau_{A_1}(A_G^{U'}) + (a, A_0) + (b, A_1)\right) \backslash \{A_0, A_1, a, b\} \right\rangle \\ \hfill \text{if } (A_0, A_1) \notin A_G^{U'} \end{cases}$$

A.3.15. $C_0 = HS, C_1 = S$

This is similar to section A.3.14.

$$\text{G.T.}^* \left(G, A_G^{U'}, HS, S\right)$$

$$\sim_{SQC} \begin{cases} \left| \tau_{A_0}\rho_{aA_0}\rho_{bA_1}\left(G + \tau_{A_0}(A_G^{U'}) + (a, A_0) + (b, A_1)\right) \backslash \{A_0, A_1, a, b\} \right\rangle \\ \hfill \text{if } (A_0, A_1) \in A_G^{U'} \\ \\ \left| \tau_{A_0}\tau_{A_1}\rho_{aA_0}\rho_{bA_1}\left(G + \tau_{A_0}(A_G^{U'}) + (a, A_0) + (b, A_1)\right) \backslash \{A_0, A_1, a, b\} \right\rangle \\ \hfill \text{if } (A_0, A_1) \notin A_G^{U'} \end{cases}$$

A.3.16. $C_0 = HS, C_1 = HS$

- Here we discuss the stabilizer state after gate teleportation for $C_0, C_1 = HS$, starting with equation 5.13.

$$\text{G.T.}^* \left(G, A_G^{U'}, HS, HS\right) = \left\langle B_1^{bA_1} \right| \left\langle B_1^{aA_0} \right| H_{A_0} S_{A_0}^{\dagger} H_{A_1} S_{A_1}^{\dagger} \left| G + A_G^{U'} \right\rangle$$

- Using $H_{A_1} S^{\dagger}_{A_1} = S_{A_1} \sqrt{-iX_{A_1}}$:

$$= \left\langle B_1^{bA_1} \middle| \left\langle B_1^{aA_0} \middle| S_{A_1} H_{A_0} S^{\dagger}_{A_0} \sqrt{-iX_{A_1}} \middle| G + A_G^{U'} \right\rangle \right.\right.$$

- Where we know that $\sqrt{-iX}$ is part of a local complementation, which leads to:

$$= \left\langle B_1^{bA_1} \middle| \left\langle B_1^{aA_0} \middle| S_{A_1} H_{A_0} S^{\dagger}_{A_0} \sqrt{-iZ}^{N_{A_1}(A_G^{U'})} \middle| G + \tau_{A_1}(A_G^{U'}) \right\rangle \right.\right.$$

- When A_0 and A_1 are neighbours in A_G^U, local complementing A_1 also results in an operation on A_0. This leads to two different cases, where in the first case we use that $S^{\dagger}_{A_0} \sqrt{-iZ_{A_0}} = \mathbb{1}$ and in the second case we exploit the fact that we can always do an extra identity operation $\mathbb{1} = S_{A_0} S^{\dagger}_{A_0}$:

$$= \begin{cases} \text{if } (A_0, A_1) \in A_G^{U'} & \left\langle B_1^{bA_1} \middle| \left\langle B_1^{aA_0} \middle| S_{A_1} H_{A_0} \sqrt{-iZ}^{N_{A_1}(A_G^{U'}) \backslash A_0} \middle| G + \tau_{A_1}(A_G^{U'}) \right\rangle \right.\right. \\ \text{if } (A_0, A_1) \notin A_G^{U'} & \left\langle B_1^{bA_1} \middle| \left\langle B_1^{aA_0} \middle| S_{A_1} \sqrt{-iZ}^{N_{A_1}(A_G^{U'})} S_{A_0} S^{\dagger}_{A_0} H_{A_0} S^{\dagger}_{A_0} \middle| G + \tau_{A_1}(A_G^{U'}) \right\rangle \right.\right. \end{cases}$$

- Using similar approaches as before in this chapter, the remaining operations can be written either as a local complementation or a pivot on the graph state, up to some SQC operations. This results in:

$$\sim_{SQC} \begin{cases} \left\langle 0_{A_0} - i_{A_1} 0_a 0_b \middle| \middle| \rho_{aA_0} \rho_{bA_1} \left(G + \rho_{A_0 u} \tau_{A_1}(A_G^{U'}) + (a, A_0) + (b, A_1) \right) \right\rangle \\ \hfill \text{if } (A_0, A_1) \in A_G^{U'} \\ \\ \left\langle -i_{A_0} - i_{A_1} 0_a 0_b \middle| \middle| \rho_{aA_0} \rho_{bA_1} \left(G + \tau_{A_0} \tau_{A_1}(A_G^{U'}) + (a, A_0) + (b, A_1) \right) \right\rangle \\ \hfill \text{if } (A_0, A_1) \notin A_G^{U'} \end{cases}$$

- Using the two qubit meaurement rules from section 5.4.2:

$$\sim_{SQC} \begin{cases} \left| \tau_{A_1} \rho_{aA_0} \rho_{bA_1} \left(G + \rho_{A_0 u} \tau_{A_1}(A_G^{U'}) + (a, A_0) + (b, A_1) \right) \backslash \{A_0, A_1, a, b\} \right\rangle \\ \hfill \text{if } (A_0, A_1) \in A_G^{U'} \\ \\ \left| \tau_{A_0} \tau_{A_1} \rho_{aA_0} \rho_{bA_1} \left(G + \tau_{A_0} \tau_{A_1}(A_G^{U'}) + (a, A_0) + (b, A_1) \right) \backslash \{A_0, A_1, a, b\} \right\rangle \\ \hfill \text{if } (A_0, A_1) \notin A_G^{U'} \end{cases}$$