University of Stuttgart

Institute for Theoretical Physics III

Bachelor thesis

Minimal instance for topological matter

Nicolai Lang

August 22, 2011

Supervised by
Prof. Dr. H.P. Büchler

Statutory Declaration

I herewith formally declare that I have written the submitted thesis independently. I did not use any outside support except for the quoted literature and other sources mentioned in the paper.

I clearly marked and separately listed all of the literature and all of the other sources which I employed when producing this academic work, either literally or in content.

Stuttgart, August 22, 2011

Nicolai Lang

Abstract

Quantum information theory is a promising interdisciplinary field of both experimental and theoretical physics on the one hand and informatics on the other. However, the storage and protection of coherent quantum information proves much more challenging than in the classical analogue. This led to various approaches in order to protect quantum information from decoherence due to inevitable interactions with the environment.

Whereas classical quantum error correction codes take the approach of *detecting* and *correcting* errors algorithmically the investigation of *topological phases* gave rise to the notion of error correction on a *physical level*. These theories use degenerate ground states of topologically ordered systems to store quantum information reliably. For topologically ordered systems the successful description of phase transitions by means of spontaneous symmetry breaking and *local* order parameters fails. This leads to an inherent stability of such phases against local noise.

A well-established theory showing topological order is *Kitaev's Toric Code Model* [1] which was subject to intensive studies in recent years. This theory describes a square lattice of spins embedded into the surface of a torus and features a topology dependent ground state degeneracy. Since the Toric Code can be described in terms of *stabilizers* – a group theoretic formalism known from quantum information theory – its ground states may be expressed as *graph states*, a subclass of multi-qubit states described by mathematical graphs and introduced by Hein *et al.* [2].

In the first part of this thesis the construction of ground states for small Toric Code systems is investigated. We show how ground states from larger systems may be derived from ground states of their smaller constituents. In the second part it is shown how a known algorithm, developed for stabilizer states in general, is applied to the special case of Toric Code Models. Furthermore a transformation rule stated in purely graph theoretic terms is provided. In the last section the existence of special, so called *local* graph states for certain Toric Code systems is investigated. We compute the smallest settings for Toric Code Models on triangular and square lattices that cannot be constructed from local graph states by means of local unitaries alone.

Zusammenfassung

Die Quanteninformationstheorie ist ein vielversprechendes, interdisziplinäres Forschungsfeld zwischen experimenteller und theoretischer Physik auf der einen und Informatik auf der anderen Seite. Im Rahmen grundlegender Forschungen hat sich herausgestellt, dass das verlässliche Speichern kohärenter Quanteninformation weitaus höhere Anforderungen stellt als man es von der klassischen Informationsverarbeitung her kannte. Dies führte zu einer Vielzahl verschiedener Ansätze um quantenmechanische Zustände vor Dekohärenz zu bewahren, die wegen unvermeidlicher Wechselwirkungen mit der Umwelt die Lebensdauer quantenmechanischer Zustände begrenzt.

Während sich klassische quantenmechanische Fehlerkorrekturcodes eines algorithmischen Ansatzes bedienen, bei dem Fehler zuerst *detektiert* und anschließend *korrigiert* werden, führte die Entdeckung *topologischer Phasen* zur Idee Fehlerkorrektur auf *physikalischer Ebene* durchzuführen. In diesen Modellen werden die entarteten Grundzustände topologisch geordneter Systeme zur zuverlässigen Speicherung von Quanteninformation benutzt. Bei topologischen Ordnungen versagen die etablierten Erklärungsansätze mittels spontaner Symmetriebrechung und lokaler Ordnungsparameter. Diese besonderen Eigenschaften solcher Phasen führen zu einer systemimmanenten Robustheit gegen lokale Störungen.

Das bekannteste Modell, das topologische Ordnung aufweist, ist *Kitaev's Toric Code* [1]. Es war in den letzten Jahren Gegenstand intensiver Forschung. Dieses Modell beschreibt die quantenmechanische Struktur eines quadratischen Gitters aus Spins auf der Oberfläche eines Torus und zeichnet sich durch eine von der Topologie abhängige Grundzustandsentartung aus. Da man den Toric Code im Rahmen des aus der Quanteninformationstheorie bekannten *Stabilizer*-Formalismus beschreiben kann, ist es möglich seine Grundzustände als *Graphenzustände* zu schreiben. Diese bilden eine Unterklasse von Multi-Qubit-Zuständen, die man mittels mathematischer Graphen beschreiben kann, und wurden von Hein *et al.* [2] eingeführt.

Im ersten Teil dieser Arbeit wird die Konstruktion der Grundzustände kleiner Toric Code Systeme untersucht. Es wird gezeigt, wie die Grundzustände größerer Systeme von den Grundzuständen ihrer Subsysteme abgeleitet werden können. Im zweiten Teil wenden wir einen für allgemeine Stabilizer-Zustände entwickelten Algorithmus auf den Spezialfall des Toric Codes an. Weiterhin geben wir eine Regel für diese Transformation an, die ausschließlich graphentheoretisch formuliert werden kann. Im letzten Abschnitt werden spezielle Toric Code Systeme auf die Existenz sogenannter *lokaler* Graphenzustände hin untersucht. In diesem Zusammenhang berechnen wir die kleinstmöglichen Spinsysteme für den Toric Code auf Dreiecks- und Quadratgittern, die nicht unter Anwendung lokal-unitärer Operationen von lokalen Graphenzuständen abgeleitet werden können.

Contents

Chapter 1

Introduction

1.1 Preliminaries

To prevent misconceptions due to different notations used by different communities some symbols, as they will be used in this thesis, are defined in the following. First, as we will deal with abstract qubits independent of their physical implementation, only the Hilbert spaces defined below will be considered.

Definition 1.1.1: (Qubit, Computational basis). *Set $\mathcal{H}_2 := \mathbb{C}^2$ and $\mathcal{H}^N := \bigotimes_{i=1}^N \mathcal{H}_2^{(i)}$. A Vector $|\Psi\rangle \in \mathcal{H}_2$ is called the **state of a qubit** or simply **qubit** identifying the **physical** qubit as a two state system with its abstract representation as a vector in $\mathcal{H}_2$. Let*

$$|0\rangle := \begin{bmatrix}1\\0\end{bmatrix} \qquad \text{and} \qquad |1\rangle := \begin{bmatrix}0\\1\end{bmatrix} \tag{1.1}$$

*be the standard basis of $\mathcal{H}_2$ (also called **computational basis**). Then an arbitrary state of a qubit is given by*

$$|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \qquad \text{where} \qquad \alpha, \beta \in \mathbb{C} \qquad \text{and} \qquad |\alpha|^2 + |\beta|^2 = 1 \tag{1.2}$$

$(\mathcal{H}_2, (\cdot,\cdot))$ becomes a Hilbert space in a natural way

$$(|\Psi_1\rangle\,, |\Psi_2\rangle) \equiv \langle \Psi_1 \,|\, \Psi_2\rangle := \alpha_1\alpha_2^* + \beta_1\beta_2^* \tag{1.3}$$

$\langle\Psi|$ denotes the linear functional $\langle\Psi| \in \mathcal{H}_2'$ where $\mathcal{H}_2'$ denotes the dual space of $\mathcal{H}_2$ and it holds $\langle\Psi|\,|\Phi\rangle = \langle\Psi \,|\, \Phi\rangle$ for all $|\Phi\rangle \in \mathcal{H}_2$. Since every linear functional $\langle\Psi|$ can be mapped to a unique vector $|\Psi\rangle$ by an isometric isomorphism provided by the Riesz representation theorem [3] *this gives rise to the famous Dirac or bra-ket notation.*

In addition to $|0\rangle$ and $|1\rangle$ the following special qubit states, which form an alternative orthonormal basis of $\mathcal{H}_2$, will be used frequently

$$|+\rangle := \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix} \qquad \text{and} \qquad |-\rangle := \frac{1}{\sqrt{2}}\begin{bmatrix}1\\-1\end{bmatrix} \tag{1.4}$$

This leads to the definition of the Pauli matrices:

Definition 1.1.2: (Pauli matrices, Pauli group). *The **Pauli matrices** are defined as*

$$\sigma^x := \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}, \quad \sigma^y := \begin{bmatrix}0 & -\mathrm{i}\\\mathrm{i} & 0\end{bmatrix}, \quad \sigma^z := \begin{bmatrix}1 & 0\\0 & -1\end{bmatrix} \tag{1.5}$$

and are identified with automorphisms acting on $\mathcal{H}_2$ by choosing the computational basis $\{|0\rangle, |1\rangle\}$. As linear operators acting on $\mathcal{H}_2$ they can be expressed by means of every dual basis. If we choose the dual basis corresponding to their eigenvectors this reads for σ^x and σ^z

$$\sigma^x = |+\rangle\langle +| - |-\rangle\langle -| \tag{1.6a}$$
$$\sigma^z = |0\rangle\langle 0| - |1\rangle\langle 1| \tag{1.6b}$$

*The **Pauli group** [4] $(G_1, \cdot)$ acting on $\mathcal{H}_2$ is defined as the set*

$$G_1 := \operatorname{span}\{\sigma^x, \sigma^y, \sigma^z\} = \{\pm\mathbb{1}, \pm i\mathbb{1}, \pm\sigma^x, \pm i\sigma^x, \pm\sigma^y, \pm i\sigma^y, \pm\sigma^z, \pm i\sigma^z\} \tag{1.7}$$

together with the matrix product $(\cdot)$. Obviously $G_1 < U(2)$ where $U(2)$ denotes the unitary group for a single qubit. The Pauli group acting on $\mathcal{H}^N$ is defined as

$$G_N := \operatorname{span}\left\{\mathbb{1}_1 \otimes \cdots \otimes \mathbb{1}_{k-1} \otimes \sigma^i_k \otimes \mathbb{1}_{k+1} \otimes \cdots \otimes \mathbb{1}_N \,|\, i \in \{x, y, z\}; 1 \leq k \leq N, k \in \mathbb{N}\right\} \tag{1.8}$$

and $G_N < U\left(2^N\right)$ thus describing certain unitary transformations on N qubits. At this point it may be appropriate to recall the important relations

$$\sigma^i\sigma^j = i\varepsilon_{ijk}\sigma^k + \delta_{ij}\mathbb{1} \tag{1.9a}$$
$$\left[\sigma^i, \sigma^j\right] = 2i\varepsilon_{ijk}\sigma^k \tag{1.9b}$$
$$\left\{\sigma^i, \sigma^j\right\} = 2\delta_{ij}\mathbb{1} \tag{1.9c}$$

where $[\cdot,\cdot]$ denotes the commutator and $\{\cdot,\cdot\}$ the anticommutator.

Besides $|0\rangle$,$|1\rangle$,$|+\rangle$ and $|-\rangle$ there are further (multi-) qubit states that are usually denoted by a distinct symbol. The following states are an orthonormal basis of $\mathcal{H}^2$ and are known as **Bell states**:

$$\left|\Phi^\pm\right\rangle := \frac{1}{\sqrt{2}}\left[|0\rangle_1 \otimes |0\rangle_2 \pm |1\rangle_1 \otimes |1\rangle_2\right] = \frac{1}{\sqrt{2}}\begin{cases}|+\rangle_1 \otimes |+\rangle_2 + |-\rangle_1 \otimes |-\rangle_2\,, & (+)\\ |+\rangle_1 \otimes |-\rangle_2 + |-\rangle_1 \otimes |+\rangle_2\,, & (-)\end{cases} \tag{1.10a}$$

$$\left|\Psi^\pm\right\rangle := \frac{1}{\sqrt{2}}\left[|0\rangle_1 \otimes |1\rangle_2 \pm |1\rangle_1 \otimes |0\rangle_2\right] = \frac{1}{\sqrt{2}}\begin{cases}|+\rangle_1 \otimes |+\rangle_2 - |-\rangle_1 \otimes |-\rangle_2\,, & (+)\\ |-\rangle_1 \otimes |+\rangle_2 - |+\rangle_1 \otimes |-\rangle_2\,, & (-)\end{cases} \tag{1.10b}$$

Consequently $\{|\Phi^\pm\rangle, |\Psi^\pm\rangle\}$ is called **Bell basis**. These four states are often used to demonstrate the physical meaning of *entanglement*, the existence of non-classical correlations that cannot be explained by any (local) hidden variable theory, that is. Note that it holds

$$\sigma^z_i\left|\Phi^\pm\right\rangle = \left|\Phi^\mp\right\rangle \tag{1.11a}$$
$$\sigma^z_i\left|\Psi^\pm\right\rangle = (-1)^{i-1}\left|\Psi^\mp\right\rangle \tag{1.11b}$$
$$\sigma^x_i\left|\Phi^+\right\rangle = \left|\Psi^+\right\rangle, \quad \sigma^x_i\left|\Phi^-\right\rangle = (-1)^i\left|\Psi^-\right\rangle \tag{1.11c}$$
$$\sigma^x_i\left|\Psi^+\right\rangle = \left|\Phi^+\right\rangle, \quad \sigma^x_i\left|\Psi^-\right\rangle = (-1)^i\left|\Phi^-\right\rangle \tag{1.11d}$$

where $i \in \{1, 2\}$. These transformations will be important later on.

The prime example for *multipartite entanglement* is given by the **GHZ states** (Greenberger-Horne-Zeilinger states)

$$|\mathrm{GHZ}_N\rangle := \frac{1}{\sqrt{N}}\left[|0\rangle^{\otimes N} + |1\rangle^{\otimes N}\right] \tag{1.12}$$

Most of the theory used to describe stabilizer states (see sec. 1.2) is based on group theory in general and especially the theory of G-sets. Therefore the next definition is presented in purely group theoretic terms and is rather important for linking theory and application than understanding what follows in physics.

Definition 1.1.3: (G-SET, STABILIZER, NORMALIZER, CENTRALIZER). *In the following G denotes a group. For simplicity the group product is omitted.*

(i) Let M be an arbitrary set. Then we call a binary mapping

$$\cdot\ :\ G \times M \to M\ :\ (g, m) \mapsto g \cdot m \tag{1.13}$$

a ***left group action*** *of G on M if $\cdot$ satisfies*

(a) $(gh) \cdot m = g \cdot (h \cdot m)$ for all $g, h \in G$ and $m \in M$.

(b) $e \cdot m = m$ for all $m \in M$ if e denotes the identity in G.

Then M is called a ***left G-set*** *or for short* ***G-set****.*

(ii) Let M be a G-set and $m \in M$. Then the ***stabilizer*** *of m is*

$$\mathrm{Stab}_G(m) := \{g \in G \,|\, g \cdot m = m\} \tag{1.14}$$

and $\mathrm{Stab}_G(m)$ is a subgroup of G.

(iii) Let $U \subseteq G$ be an arbitrary subset. Then we call

$$N_G(U) := \left\{g \in G \,|\, gUg^{-1} = U\right\} \tag{1.15}$$

the ***normalizer*** *of U in G and $N_G(U) \leq G$.*

(iv) Let $U \subseteq G$ be an arbitrary subset. Then we call

$$C_G(U) := \{g \in G \,|\, \forall_{u \in U}\ :\ gu = ug\} \tag{1.16}$$

the ***centralizer*** *of U in G and $C_G(U) \leq G$. Especially $Z(G) := C_G(G)$ is called the* ***center*** *of G and is by definition an abelian subgroup of G.*

In our case the Pauli group G_1 acts on the states of one qubit, i.e. $\mathcal{H}^1$, if we identify the matrices of G_1 with their automorphisms on $\mathcal{H}^1$ given by the computational basis $\{|0\rangle, |1\rangle\}$. Therefore $\mathcal{H}^1$ is a G_1-set in group theoretic terms. This applies analogously to the case of N qubits described by $\mathcal{H}^N$ and G_N.

In the field of quantum error correction and multipartite entanglement it is of great importance if a given unitary operation $u \in U(2^N)$ leaves G_N invariant under conjugation. This leads to the following

Definition 1.1.4: (CLIFFORD GROUP, LOCAL CLIFFORD GROUP). *The* ***Clifford group*** *$\mathcal{C}_N$ is defined as the normalizer of G_N, i.e.*

$$\mathcal{C}_N := N_{U(2^N)}\left(G_N\right) = \left\{u \in U\left(2^N\right) \,|\, uG_Nu^\dagger = G_N\right\} \tag{1.17}$$

Especially for a single qubit this reads

$$\mathcal{C}_1 = \left\{u \in U\left(2\right) \,|\, uG_1u^\dagger = G_1\right\} \tag{1.18}$$

Furthermore the ***local Clifford group*** [5] *$\mathcal{C}_N^l$ is defined as*

$$\mathcal{C}_N^l := \{c_1 \otimes \cdots \otimes c_N \,|\, c_i \in \mathcal{C}_1, 1 \leq i \leq N\} = \mathcal{C}_1^{\otimes N} \tag{1.19}$$

and consequently $\mathcal{C}_N^l \leq \mathcal{C}_N$.

The local Clifford group will be of great importance in chapters 3 and 4.

1.2 The stabilizer formalism

1.2.1 Definition and properties

We are going to introduce a formalism widely used in quantum information theory to describe quantum states and transformations of the latter in a more efficient way than by means of linear operators acting directly on state vectors. This formalism is called the *stabilizer formalism* which is an application of G-set theory known from algebra. The following introduction focuses on the relevant notions used later on and skips proofs and detailed analysis. These and further explanations may be found in [4] where a whole chapter is dedicated to stabilizer codes.

Definition 1.2.1: (STABILIZER). *Let $\mathcal{H}^N$ be the state space of N qubits and G_N the Pauli group operating on their states. Suppose that $\mathcal{G} = \{g_i\}_{i=1}^{d} \subseteq G_N$ is a given set of independent, pairwise commuting Pauli operators. Then we call*

$$\mathcal{S} := \operatorname{span} \mathcal{G} \tag{1.20}$$

the ***stabilizer*** *of the protected subspace*

$$\mathcal{PS} := \left\{ |\Phi\rangle \in \mathcal{H}^N \,|\, \mathcal{S} \,|\Phi\rangle = |\Phi\rangle \right\} \tag{1.21}$$

and $\mathcal{G}$ the ***generators*** *of $\mathcal{S}$. $d = \operatorname{rank} \mathcal{S}$ is called the rank of the stabilizer. Note that $\mathcal{S}$ is an abelian subgroup of G_N by definition.*

The definition above starts with the generators $\mathcal{G}$ and subsequently constructs the protected space $\mathcal{PS}$. In this case it may happen, that the protected space is degenerate, i.e. $\dim \mathcal{PS} > 1$ or even trivial. One can show [4] that $-\mathbb{1} \notin \mathcal{S}$ is a necessary condition for $\mathcal{PS}$ being non-trivial. To this end let us state the following proposition

Proposition 1.2.2: (DIMENSION OF $\mathcal{PS}$). *Let $\mathcal{G}$ be the independent generators of a stabilizer $\mathcal{S}$ such that $-\mathbb{1} \notin \mathcal{S}$ acting on $\mathcal{H}^N$ with $|\mathcal{G}| = d$. Then it follows*

$$\dim \mathcal{PS} = 2^{N-d} \tag{1.22}$$

The proof can be found in [4]. Proposition 1.2.2 will be used throughout this thesis and should be kept in mind.

In case of $\dim \mathcal{PS} = 1$ the link to def. 1.1.3 is established by providing $|\Psi\rangle \neq 0$ as protected state. Hence $\mathcal{PS} := \operatorname{span} \{|\Psi\rangle\}$ and it follows $\mathcal{S} = \operatorname{Stab}_{G_N}(|\Psi\rangle)$ for the corresponding stabilizer, which turns out to be an abelian subgroup of G_N since elements of G_N commute or anti-commute and $|\Psi\rangle \neq 0$. By Proposition 1.2.2 we deduce $N = d$ and therefore we have $\mathcal{G} = \{g_i\}_{i=1}^{N}$ for appropriately chosen g_i.

Up to now we just found a new way defining N-qubit states in terms of stabilizer operators. This gives rise to two questions:

1. What conditions hold for an operator transforming stabilizer states into stabilizer states?
2. How may such a transformation be described in terms of stabilizers rather than in terms of state vectors?

To this end suppose that $|\Psi\rangle \in \mathcal{PS}$ is stabilized by $\mathcal{S}$ and transformed by a unitary $U \in U\left(2^N\right)$. Let

$$U\mathcal{S}U^\dagger := \left\{ UsU^\dagger \,|\, s \in \mathcal{S} \right\} \tag{1.23}$$

be the transformed stabilizer. Obviously $U\mathcal{S}U^\dagger \leq U\left(2^N\right)$ is an abelian subgroup of the unitary group acting on N spins. Then it holds

$$UsU^\dagger U \,|\Psi\rangle = Us \,|\Psi\rangle = U \,|\Psi\rangle \tag{1.24}$$

and $|\Psi'\rangle = U|\Psi\rangle$ is stabilized by $\mathcal{S}' = U\mathcal{S}U^\dagger$. In general $\mathcal{PS}' = U\mathcal{PS}$ is stabilized by $\mathcal{S}' = U\mathcal{S}U^\dagger$ and $\mathcal{S}' = \operatorname{span}\mathcal{G}'$ where $\mathcal{G}' = \left\{Ug_iU^\dagger\right\}_{i=1}^{d}$. This observation answers question two: One may transform stabilizer states by conjugation of the generators with the unitaries in question. This is an efficient way to keep track of the states without dealing with high dimensional vector spaces, e.g. $\mathcal{H}^N$ (see Theorem 1.2.3 below). For example, let σ^x be a generator and consider the Hadamard transformation

$$H := \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \tag{1.25}$$

Then one obtains easily $H\sigma^xH^\dagger = \sigma^z$ and of course $H\sigma^zH^\dagger = \sigma^x$. This leads to the core of question one. Obviously $H\sigma^xH^\dagger = \sigma^z \in G_N$ (identities acting on other spins are omitted) and one can show that indeed $H\mathcal{S}H^\dagger \leq G_N$. According to def. 1.2.1 a stabilizer $\mathcal{S}$ is a subgroup of the *Pauli group* G_N. For arbitrary $U \in U\left(2^N\right)$ this is not the case. The condition we are looking for is $U \in \mathcal{C}_N$ since then $UG_NU^\dagger = G_N$ and $U\mathcal{S}U^\dagger \leq G_N$ is a new stabilizer in the sense of def. 1.2.1.

Finally one may ask, if it is advantageous to use the stabilizer formalism for the description of quantum mechanical state transformations. To this end we cite the Gottesman-Knill theorem as stated in [4].

Theorem 1.2.3: (GOTTESMAN-KNILL THEOREM). *A quantum computation which involves only the following elements*

- *State preparation in the computational basis.*
- *Hadamard gates, phase gates, C-NOT gates and Pauli gates.*
- *Measurements of observables in the Pauli group.*
- *Classical control of these gate operations conditioned on the outcome of such measurements.*

may be efficiently simulated on a classical computer.

Remark 1.2.4. *Here* ***efficiently*** *means, that the computation time needed to simulate a system of N qubits running a quantum code with M of these special operations grows polynomially with N and M. Therefore the corresponding problem belongs to the complexity class* **P**.

According to the explanations given above, stabilizer codes belong to the class of quantum mechanical state transformations required by Theorem 1.2.3. Therefore any stabilizer circuit may be simulated efficiently on a classical computer. This is a remarkable result since a common view of quantum computing propagates the notion of quantum computers being "faster" than classical computers. That this holds not true for arbitrary quantum codes is shown by Theorem 1.2.3[1].

1.2.2 Binary representation of stabilizer codes

Often – as we will see later – it is not important to keep track of the phase factors of each stabilizer generator, since states described by stabilizers differing only in such local phases are equivalent under local unitary transformations (for LU-equivalence see sec. 1.7). Therefore the following representation of a given stabilizer is useful if we are only interested in LU-invariant properties of stabilizer states:

Definition 1.2.5: (BINARY REPRESENTATION, GENERATOR MATRIX). *Let $\mathbb{F}_2$ be the field over two elements*[2]*, i.e. the set $\{0,1\}$ with addition modulo 2 and multiplication as known from $\mathbb{R}$. Let $\mathcal{S}$ be a stabilizer with N generators $\mathcal{G} = \{g_i\}_{i=1}^{N} \subseteq G_N$. Since $\mathcal{I} := \operatorname{span}\{\pm\mathbb{1}, \pm i\mathbb{1}\}$ is a normal subgroup of G_N its intersection with $\mathcal{S}$ is a normal subgroup with respect to $\mathcal{S}$. Therefore we may factor out and get the reduced stabilizer*

$$\mathcal{S}^* := {}^{\mathcal{S}}/_{\mathcal{I}} \tag{1.26}$$

[1] It turns out to be exceedingly difficult finding quantum algorithms that outpace their classical analogues [6,7].

[2] Such finite fields are called *Galois fields* GF(n), i.e. $\mathbb{F}_2 = \mathrm{GF}(2)$.

In this factor group computations are performed as in $\mathcal{S}$ ignoring the phase factors. For the sake of simplicity embed $\mathcal{S}$ into $\mathcal{S}^*$ by means of the natural surjection and write $\mathcal{G}$ for a set of representatives generating $\mathcal{S}^*$. From this point of view w.l.o.g. g_i has the form

$$g_i = \sigma_1^{\nu_1} \otimes \cdots \otimes \sigma_N^{\nu_N} \quad \textit{where} \quad \nu_i \in \{x, y, z, 0\} \quad \textit{and} \quad \sigma^0 := \mathbb{1} \tag{1.27}$$

Now define a group homomorphism $\delta : \mathcal{S}^* \to \mathbb{F}_2^{2N}$ by

$$\delta\left(g_i\right)_j := \begin{cases} 1 & j \leq N : \nu_j = z \vee \nu_j = y \,;\, j > N : \nu_{j-N} = x \vee \nu_{j-N} = y \\ 0 & \textit{otherwise} \end{cases} \tag{1.28}$$

The first N entries of $\delta\left(g_i\right)$ indicate, if σ^z or $\mathbb{1}$ operate on the corresponding qubit whereas the last N entries contain information about the action of σ^x operators. One easily confirms that δ maps $\mathcal{S}^*$ homeomorphic into $\mathbb{F}_2^{2N}$ (viewed as an additive group) and is well defined. In fact, it is a monomorphism of groups, i.e. injective. Therefore it becomes a group isomorphism if we restrict the range to $\delta\mathcal{S}^* = \mathrm{im}(\delta)$ (note that for injectivity we need the quotient $\mathcal{S}^*$ as domain). The generating set of $\delta\mathcal{S}^*$ can be written as a ***generator matrix***

$$\mathbf{M}\left[\mathcal{G}\right] := \left[\delta\left(g_1\right), \ldots, \delta\left(g_N\right)\right] = \begin{bmatrix} \mathbf{Z} \\ \mathbf{X} \end{bmatrix} = \left[\begin{array}{ccc} \delta\left(g_1\right)_1 & \cdots & \delta\left(g_N\right)_1 \\ \vdots & \ddots & \vdots \\ \delta\left(g_1\right)_N & \cdots & \delta\left(g_N\right)_N \\ \hline \delta\left(g_1\right)_{N+1} & \cdots & \delta\left(g_N\right)_{N+1} \\ \vdots & \ddots & \vdots \\ \delta\left(g_1\right)_{2N} & \cdots & \delta\left(g_N\right)_{2N} \end{array}\right] \tag{1.29}$$

with $N \times N$-submatrices $\mathbf{Z}$ and $\mathbf{X}$.

The subspace $\delta\mathcal{S}^* \leq \mathbb{F}_2^{2N}$ as a vector space over $\mathbb{F}_2$ shows some interesting properties [4,5,8]. In the first place we define a *symplectic structure* on $\mathbb{F}_2^{2N}$. The matrix

$$\mathbf{P} := \begin{bmatrix} \mathbf{0} & \mathbf{E}_N \\ \mathbf{E}_N & \mathbf{0} \end{bmatrix} \tag{1.30}$$

(here $\mathbf{E}_N$ denotes the identity matrix in $\mathbb{F}_2^N$) induces the *symplectic inner product*, i.e. a totally isotropic and non-degenerate bilinear form

$$\langle \mathbf{a}, \mathbf{b} \rangle := \mathbf{a}^T \mathbf{P} \mathbf{b} \quad \text{where} \quad \mathbf{a}, \mathbf{b} \in \mathbb{F}_2^{2N} \tag{1.31}$$

By straightforward calculation one finds, that two operators $A, B \in G_N$ commute iff they are orthogonal with respect to this symplectic inner product. Formally this reads

$$[A, B] = 0 \Leftrightarrow \langle \delta(A), \delta(B) \rangle = 0 \tag{1.32}$$

$\mathbf{M}\left[\mathcal{G}\right]$ is called *generator matrix* of the the stabilizer $\mathcal{S}^*$ since for all $A \in \mathcal{S}^*$ there exists a binary vector $\mathbf{x} \in \mathbb{F}_2^N$ such that $\delta(A) = \mathbf{M}\left[\mathcal{G}\right]\mathbf{x}$. Often $\mathcal{S}$ is written instead of $\mathcal{S}^*$ and neglecting the phases is implicitly assumed.

By definition of a stabilizer $\mathcal{S}$ it holds

$$\mathbf{M}\left[\mathcal{G}\right]^T \mathbf{P} \mathbf{M}\left[\mathcal{G}\right] = \mathbf{0} \tag{1.33}$$

since all generators commute. Hence one calls the subspace $\delta\mathcal{S}^*$ *self-orthogonal* with respect to the symplectic product. Furthermore it turns out, that a set of generating Pauli operators $\mathcal{G} \subset G_N$ $(-\mathbb{1} \notin \mathcal{G})$ is independent iff the generator matrix has maximal rank, i.e. $\operatorname{rank} \mathbf{M}\left[\mathcal{G}\right] = N$ [4]. This statement is quite useful since one easily checks the latter condition by applying the machinery of linear algebra instead of checking the (in-)dependence of Pauli operators directly. In this picture a change of the basis of $\mathcal{S}^*$ (i.e. transforming the independent set $\mathcal{G}$ of generators to a new independent set $\mathcal{G}'$ by concatenation of operators) is achieved by multiplication of the generator matrix with an invertible $N \times N$-matrix from the right. This reads formally

$$\mathbf{M}\left[\mathcal{G}'\right] = \mathbf{M}\left[\mathcal{G}\right]\mathbf{T}\left[\mathcal{G},\mathcal{G}'\right] \quad \text{where} \quad \mathbf{T}\left[\mathcal{G},\mathcal{G}'\right] \in \mathrm{GL}\left(N,\mathbb{F}_2\right) \tag{1.34}$$

At last we give the following definition of a special type of generator matrices which will become important later on (see sec. 1.6).

Definition 1.2.6: (Adjacency matrix, Graph state generator). *Let* $\mathbf{\Gamma} \in \mathbb{F}_2^{N\times N}$ *be a matrix with the following properties*

1. $\mathbf{\Gamma} = \mathbf{\Gamma}^T$*, i.e. the matrix is symmetric.*

2. $\Gamma_{ii} = 0$ *for all* $i \in \{1,\dots,N\}$*.*

Then we call $\mathbf{\Gamma}$ *an **adjacency matrix** and the corresponding generator matrix of a stabilizer* $\mathcal{S}\left[\mathbf{\Gamma}\right]$

$$\mathbf{M}\left[\mathbf{\Gamma}\right] := \begin{bmatrix} \mathbf{\Gamma} \\ \mathbf{E}_N \end{bmatrix} \tag{1.35}$$

*is called **graph state generator**.*

1.3 A toy model: The Toric Code

In 2002 Kitaev proposed a spin model called the *Toric Code Model* [1] (TCM) showing the inherent capability of quantum error correction on a physical level. Due to the No-cloning theorem the correction of errors to conserve *coherent* states in the presence of noise is a subtle matter and requires more sophisticated methods than classical error correction algorithms. A second (and not independent) property of the TCM is the presence of a so called *topological phase* at $T = 0$ that cannot be described by Landau theory dealing with second-order phase transitions described by *local* order parameters.

Up to now many other theories showing topological order have been discovered [9–11], but the TCM is the most prominent and simplest one featuring interesting properties. Therefore it can be considered a toy model of topological order showing that this new kind of order may be used for fault tolerant quantum computing (in this context called *topological quantum computing*). Let us now introduce the Toric Code Model and investigate its properties.

1.3.1 The setting

Let $\mathcal{L}$ be a square lattice with N cells in each direction. We denote the vertices, edges and faces of the lattice by s, i and p respectively. $V(\mathcal{L})$, $E(\mathcal{L})$ and $P(\mathcal{L})$ denote the sets of vertices, edges and faces of $\mathcal{L}$. Embed this lattice into the oriented 2-manifold with genus $g = 1$ (see sec. 1.4.1) – also known as *torus* – as shown in fig. 1.1. Here the upper border is identified with the (dashed) lower border and the left border with the (dashed) right border. Attach a single spin (or qubit) to each of the $2N^2$ edges of the lattice described by state vectors in $\mathcal{H} = \bigotimes_{i\in E(\mathcal{L})} \mathcal{H}_2^{(i)}$. Here $\mathcal{H}_2^{(i)}$ is the Hilbert space describing the spin on edge i.

Define the following operators for each vertex $s \in V(\mathcal{L})$ and face $p \in P(\mathcal{L})$

$$A_s := \bigotimes_{i\in s} \sigma_i^x \qquad \text{and} \qquad B_p := \bigotimes_{i\in p} \sigma_i^z \tag{1.36}$$

where $\sigma_i^{x,z}$ are the Pauli spin operators acting on the Hilbert space $\mathcal{H}_2^{(i)}$. The vertices (sites) s are identified with their adjacent edges $i \in s$ and thus $\mathrm{supp}\left(A_s\right)$ can be illustrated as the *star* with center s. Analogously one identifies a face p with its bounding edges i. Therefore $\mathrm{supp}\left(B_p\right)$

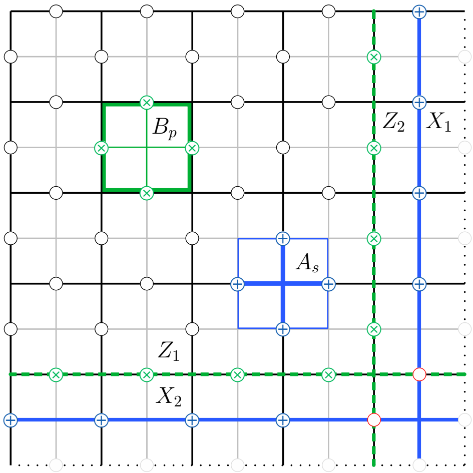


FIGURE 1.1 (Color online) : The setting used for the Toric Code Model as proposed in ref. [1]. Spins are labeled with circles on the edges of a square lattice which is embedded into the torus. Thus opposing boundaries have to be identified (actually, the grey spins are identified with their black counterparts). The lattice $\mathcal{L}$ is drawn with black lines whereas the dual lattice $\mathcal{L}^*$ appears as a thin, grey net. The star operators A_s can be seen as the plaquette operators B_s^* of the dual lattice and the plaquette operators B_p as the dual star operators A_p^*, respectively. The non-trivial loops $\mathcal{C}_{1,2}^*$ ($\mathcal{C}_{1,2}$) and their corresponding loop operators X_i (Z_i) are represented by blue (green) lines.

is illustrated as the *plaquette* filling the face p (see fig. 1.1). The A_s will be called *star operators* whereas the B_p are called *plaquette operators.*

It is easy to see that

$$[A_s, A_{s'}] = 0, \qquad [B_p, B_{p'}] = 0, \qquad [A_s, B_p] = 0 \tag{1.37}$$

for all $s, s' \in V(\mathcal{L})$ and $p, p' \in P(\mathcal{L})$. $[A_s, B_p] = 0$ follows since $|\text{supp}\,(A_s) \cap \text{supp}\,(B_p)| \in \{0, 2\}$ and $\{\sigma^x, \sigma^z\} = 0$. Furthermore the star and plaquette operators are in the *Pauli group* G_{2N^2} and consequently

$$\mathcal{S} \equiv \mathcal{S}(N) := \text{span}\,\mathcal{B} \tag{1.38}$$

is a stabilizer group where $\mathcal{B} := \{A_s, B_p\}_{s\in V(\mathcal{L}), p\in P(\mathcal{L})}$ is a generating set of $\mathcal{S}$. It is important to realize that $\mathcal{B}$ is *not* an operator basis, i.e. the star and plaquette operators are *not* independent. In fact, since every edge i is adjacent to two vertices s and two faces p (torus!) it holds

$$\prod_{s\in V(\mathcal{L})} A_s = \mathbb{1} \qquad \text{and} \qquad \prod_{p\in P(\mathcal{L})} B_p = \mathbb{1} \tag{1.39}$$

Since $|V(\mathcal{L})| = N^2 = |P(\mathcal{L})|$ and $|E(\mathcal{L})| = 2N^2$ it follows $\text{rank}\,\mathcal{S}(N) = 2N^2 - 2$ as the rank of the TCM stabilizer (the -2 is due to the two constraints in eq. (1.39)). According to Proposition 1.2.2 we have $\dim \mathcal{PS} = 2^{2N^2-(2N^2-2)} = 2^2 = 4$. Therefore the TCM yields a 4 dimensional protected subspace (also called *code space*). In section 1.4.2 we will derive, that the dimension (or degeneracy if we introduce a Hamiltonian) depends only on the genus[3] g of the surface the lattice is embedded into.

We are now going to calculate a basis of $\mathcal{PS}$. To this end we need to establish the notion of a *dual lattice* $\mathcal{L}^*$ which is drawn in fig. 1.1 with thin, grey lines. Consider $\mathcal{L}$ as a graph then $\mathcal{L}^*$ is simply the dual graph in the usual sense of graph theory, therefore consisting of vertices s^* for every face p of $\mathcal{L}$ linked by edges i^* if the corresponding faces p are adjacent. For further explanations see section 1.4.2.

Remark 1.3.1. *In the picture of the dual lattice $\mathcal{L}^*$ one may call A_s plaquette operators and B_p star operators since the duality leads to the identification $s \leftrightarrow p^*$ and $p \leftrightarrow s^*$. This is illustrated in fig. 1.1.*

In the following $\mathcal{C}_{pq}$ denotes a path from vertex p to q on the lattice $\mathcal{L}$ while $\mathcal{C}_j$ denotes a closed path with index j. Paths on the dual lattice $\mathcal{L}^*$ will be labeled by *. Define two types of *string*

[3] The number of “holes”.

operators

$$X\left[\mathcal{C}^*_{pq}\right] := \bigotimes_{i^*\in\mathcal{C}^*_{pq}} \sigma^x_{i^*} \tag{1.40a}$$

$$Z\left[\mathcal{C}_{pq}\right] := \bigotimes_{i\in\mathcal{C}_{pq}} \sigma^z_{i} \tag{1.40b}$$

where the paths $\mathcal{C}_{pq}$ and $\mathcal{C}^*_{pq}$ are identified with their edges. Let $\mathcal{C}_{1,2}$ denote two non-homologous and non-trivial paths, each of them winding around the torus as shown in fig. 1.1. For a short introduction to algebraic topology see section 1.4.1. Analogously choose $\mathcal{C}^*_{1,2}$ on the dual lattice. Then we define the *loop operators*

$$X_j := X\left[\mathcal{C}^*_j\right] \tag{1.41a}$$

$$Z_j := Z\left[\mathcal{C}_j\right] \tag{1.41b}$$

where $j \in \{1,2\}$. (in general $j \in \{1,2,\ldots,2g\}$ on an orientable surface with genus g). For an example of loop operators defined by paths homologous to 0 see fig. 1.2. Such operators act trivially on $\mathcal{PS}$ for they can be constructed from stabilizer operators A_s and B_p. This is not the case if their paths wind once around the torus.

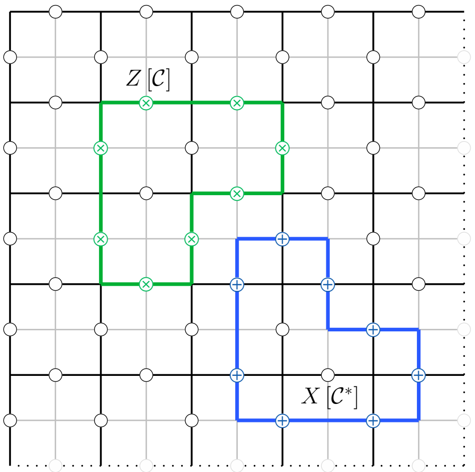


FIGURE 1.2 (Color online) : Two paths homologous to 0. The green (blue) one lives one the lattice $\mathcal{L}$ (dual Lattice $\mathcal{L}^*$) and defines a z-type (x-type) loop operator. Both act as the identity on $\mathcal{PS}$ since they can be constructed from stabilizer operators B_p and A_s, respectively.

Note that

$$[A_s, X_j] = 0, \qquad [B_p, X_j] = 0, \qquad [A_s, Z_j] = 0, \qquad [B_p, Z_j] = 0 \tag{1.42}$$

since the paths are closed, hence $|\mathrm{supp}\,(A_s) \cap \mathrm{supp}\,(Z_j)| \in \{0,2,4\}$ and $|\mathrm{supp}\,(B_p) \cap \mathrm{supp}\,(X_j)| \in \{0,2,4\}$. Furthermore we find the (anti-) commutation relations

$$[Z_i, X_j] = 0, \qquad \{Z_j, X_j\} = 0, \qquad [X_i, X_j] = 0, \qquad [Z_i, Z_j] = 0, \tag{1.43}$$

where $i \neq j$. $\{Z_j, X_j\} = 0$ follows since $\mathcal{C}_j$ and $\mathcal{C}^*_j$ intersect at one spin (marked with a red circle in fig. 1.1).

We are now ready to calculate the TCM ground states $|v_1, v_2\rangle$ where $v_i \in \{1,-1\}$ are called *topological quantum numbers*. The reason for this name will become clear later on.

Lemma 1.3.2. *Let* $|\mathrm{SEC}(v_1,v_2)\rangle \in \mathcal{H}$ *be any vector fulfilling the conditions* $B_p\,|\mathrm{SEC}(v_1,v_2)\rangle = |\mathrm{SEC}(v_1,v_2)\rangle$ *for all* $p \in P(\mathcal{L})$ *and* $Z_j\,|\mathrm{SEC}(v_1,v_2)\rangle = v_j\,|\mathrm{SEC}(v_1,v_2)\rangle$ *where* $v_j \in \{1,-1\}$. *Suppose* $\mathcal{A} := \mathrm{span}\,\{A_s\}_{s\in V(\mathcal{L})}$, *then*

$$|v_1, v_2\rangle = \frac{1}{\sqrt{2^{N^2-1}}} \sum_{A\in\mathcal{A}} A\,|\mathrm{SEC}(v_1,v_2)\rangle \tag{1.44}$$

is a (basis) vector of $\mathcal{PS}$ and

$$Z_j \left|v_1, v_2\right\rangle = v_j \left|v_1, v_2\right\rangle \tag{1.45}$$

as well as

$$X_j \left|v_1, v_2\right\rangle = \begin{cases} \left|-v_1, v_2\right\rangle, & j = 1 \\ \left|v_1, -v_2\right\rangle, & j = 2 \end{cases} \tag{1.46}$$

Therefore $\{\left|v_1, v_2\right\rangle\}_{v_i \in \{1,-1\}}$ is a ONB of $\mathcal{PS}$.

Proof. That $\left|v_1, v_2\right\rangle \in \mathcal{PS}$ follows by straightforward calculation. Since $[A, B_p] = 0$ for all $A \in \mathcal{A}$ and $p \in P(\mathcal{L})$ it follows immediately $B_p \left|v_1, v_2\right\rangle = \left|v_1, v_2\right\rangle$. Let $A_s \in \mathcal{A}$ be an arbitrary star operator. Then

$$A_s \left|v_1, v_2\right\rangle = \frac{1}{\sqrt{2^{N^2-1}}} \sum_{A \in \mathcal{A}} A_s A \left|\mathrm{SEC}(v_1, v_2)\right\rangle = \frac{1}{\sqrt{2^{N^2-1}}} \sum_{A' \in \mathcal{A}} A' \left|\mathrm{SEC}(v_1, v_2)\right\rangle = \left|v_1, v_2\right\rangle$$

Here was used that $\mathcal{A}$ is a group by definition with order $|\mathcal{A}| = 2^{N^2-1}$. Therefore $\mathcal{S} \left|v_1, v_2\right\rangle = \left|v_1, v_2\right\rangle$. Furthermore

$$Z_j \left|v_1, v_2\right\rangle = \frac{1}{\sqrt{2^{N^2-1}}} \sum_{A \in \mathcal{A}} Z_j A \left|\mathrm{SEC}(v_1, v_2)\right\rangle = \frac{1}{\sqrt{2^{N^2-1}}} \sum_{A \in \mathcal{A}} A Z_j \left|\mathrm{SEC}(v_1, v_2)\right\rangle = v_j \left|v_1, v_2\right\rangle$$

Note that Z_j is Hermitian and unitary. Therefore $\langle v_1, v_2 \,|\, v_1', v_2' \rangle = \delta_{v_1, v_1'} \delta_{v_2, v_2'}$ and $\{\left|v_1, v_2\right\rangle\}_{v_i \in \{1,-1\}}$ is a ONB of $\mathcal{PS}$ since $\dim \mathcal{PS} = 4$. Lastly consider $X_j \left|v_1, v_2\right\rangle$ and use that $\{Z_j, X_j\} = 0$. We find $(i \neq j)$

$$\begin{aligned} Z_j X_j \left|v_1, v_2\right\rangle &= -X_j Z_j \left|v_1, v_2\right\rangle = -v_j X_j \left|v_1, v_2\right\rangle \\ Z_i X_j \left|v_1, v_2\right\rangle &= X_j Z_i \left|v_1, v_2\right\rangle = v_i X_j \left|v_1, v_2\right\rangle \end{aligned}$$

This completes the proof. ■

This lemma delivers some insight into the structure of $\mathcal{PS}$. A question remaining is how we can choose the reference states $\left|\mathrm{SEC}(v_1, v_2)\right\rangle \in \mathcal{H}$. First note that these states are not unique since we may choose every state $\left|\mathrm{SEC}'(v_1, v_2)\right\rangle$ obtained from $\left|\mathrm{SEC}(v_1, v_2)\right\rangle$ by applying an arbitrary $A \in \mathcal{A}$. Secondly, since we know how to get $\left|v_1, v_2\right\rangle$ with arbitrary quantum numbers from $\left|1, 1\right\rangle$ by applying X_j, it is sufficient to know $\left|\mathrm{SEC}(1, 1)\right\rangle$. The simplest choice is $\left|\mathrm{SEC}(1, 1)\right\rangle = \left|0\right\rangle^{\otimes 2N^2}$ as one verifies easily. The reader might ask, where the label "SEC" comes from. The reason for this labeling is that $\left|\mathrm{SEC}(v_1, v_2)\right\rangle \in \mathrm{SEC}(v_1, v_2)$ where $\mathrm{SEC}(v_1, v_2)$ is a linear subspace of $\mathcal{H}$ and is called the *topological SECtor* specified by (v_1, v_2). It is defined as follows

$$\mathrm{SEC}(v_1, v_2) := \mathcal{PF}(v_1, v_2) \tag{1.47}$$

where $\mathcal{F}(v_1, v_2) := \operatorname{span}\{B_p, v_1 Z_1, v_2 Z_2\}_{p \in P(\mathcal{L})}$ is a stabilizer. Obviously we may choose for $\left|\mathrm{SEC}(v_1, v_2)\right\rangle$ an arbitrary vector from the topological sector $\mathrm{SEC}(v_1, v_2)$.

1.3.2 Excitations

Up to now the TCM was purely described in terms of stabilizers and interpreted as an abstract quantum code. Let us now implement this code by writing down the Hamiltonian

$$H_{TCM} := -\sum_{s \in V(\mathcal{L})} A_s - \sum_{p \in P(\mathcal{L})} B_p \tag{1.48}$$

Clearly every $\left|\Psi\right\rangle \in \mathcal{PS}$ is an eigenstate of H_{TCM} with energy $E_0 = -2N^2$. Since all A_s and B_p pairwise commute they are diagonalizable simultaneously with eigenvalues ±1. Therefore $\mathcal{PS}$ is the 4-fold degenerate ground state space of H_{TCM}. Excited states violate the stabilizer conditions $A_s |\Psi\rangle = |\Psi\rangle$ and $B_p |\Psi\rangle = |\Psi\rangle$ for at least one vertex s or plaquette p. Consequently the energy of the first excited states is $E^* = E_0 + 4$ and the spectrum is gapped. Remember that $\prod_{s\in V(\mathcal{L})} A_s = \mathbb{1} = \prod_{p\in P(\mathcal{L})} B_p$. If an excited state $|\Psi\rangle$ violates a star operator condition there must be another vertex where a second violation occurs. The same argument holds true for violation of plaquette conditions. It is advantageous looking at vertices or plaquettes with violated stabilizer condition as being occupied by a quasiparticle:

Definition 1.3.3: (ELEMENTARY EXCITATIONS). *Let $s \in V(\mathcal{L})$ be a vertex with $A_s |\Psi\rangle = -|\Psi\rangle$ and $p \in P(\mathcal{L})$ be a plaquette with $B_p |\Psi\rangle = -|\Psi\rangle$. Then we say that p is occupied by a **x-type particle** m and s by a **z-type particle** e.*

For reasons explained below these particles are called **anyons**. As derived above it follows that excitations in the TCM always exhibit an *even* number of anyons.

This fact corresponds to the observation that $\left\{X\left[\mathcal{C}^*_{pq}\right], B_{p,q}\right\} = 0$ and $\left[X\left[\mathcal{C}^*_{pq}\right], B_{p'}\right] = 0$ as well as $\{Z\left[\mathcal{C}_{sf}\right], A_{s,f}\} = 0$ and $[Z\left[\mathcal{C}_{sf}\right], A_{s'}] = 0$ where $p\prime \neq p, q$ and $s' \neq s, f$. In other words: If we apply a x-type *string* operator (not *loop* operator!) $X\left[\mathcal{C}^*_{pq}\right]$ to a Toric Code ground state $|\Psi\rangle \in \mathcal{PS}$ there are x-type particles at its endpoints p and q. Analogously z-type *string* operators lead to z-type particles attached to their ends.

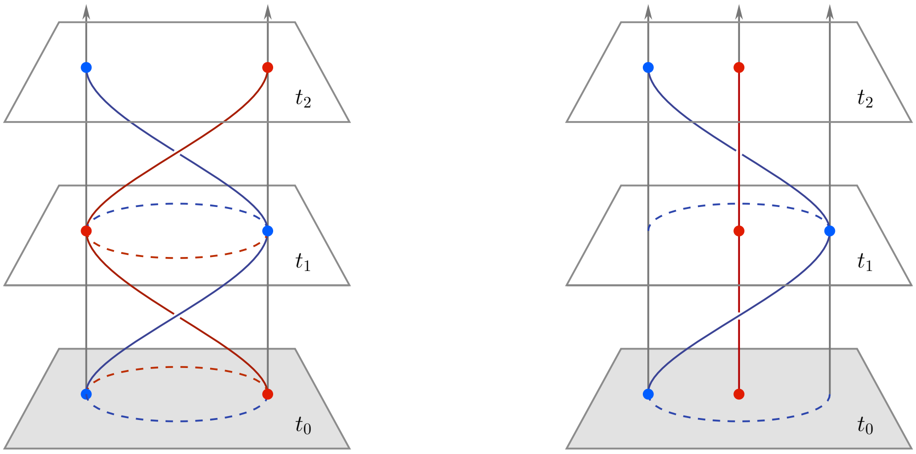


FIGURE 1.3 (Color online) : Exchanging two particles (blue and red dots) twice in the plane (left picture) is topologically equivalent to one particle winding once around the other (right picture), if continuous deformation of the trajectories in the $2 + 1$-dimensional spacetime is allowed.

Now imagine that a pair of x-type particles has been created locally (e.g. by thermal fluctuations). This state can be generated by applying $X\left[\mathcal{C}^*_{pq,1}\right]$ to a TCM ground state characterized by topological quantum numbers v_1 and v_2. "Pull" one of the particles around the torus by applying another x-type string operator $X\left[\mathcal{C}^*_{pq,2}\right]$ to the excited state where $\mathcal{C}^*_{pq,1} + \mathcal{C}^*_{pq,2} = \mathcal{C}^*_i$ is a homologous non-trivial path. What happened? We applied $X\left[\mathcal{C}^*_{pq,1}\right] X\left[\mathcal{C}^*_{pq,2}\right] = X\left[\mathcal{C}^*_{pq,1} + \mathcal{C}^*_{pq,2}\right] = X\left[\mathcal{C}^*_i\right] = X_i$ to $|v_1, v_2\rangle$ and thus flipped v_i! The new state after creating a pair of x-type anyons, pulling one around the torus and annihilating it with the other is not an excited state but another ground state with one flipped quantum number v_i.

As it is widely known, there are two types of particles obeying different statistics. They are called *fermions* if exchanging two (indistinguishable) particles yields a global phase factor $e^{i\pi} = -1$ and *bosons* if the same procedure acts as the identity on the state space, i.e. $e^{i0} = 1$. Obviously exchanging two particles twice yields in both cases the original state without any phase factor. The double exchange of two particles is topologically equivalent to one particle winding once around the other, which stays fixed. This is illustrated in fig. 1.3.

We may now ask, if there are particles obeying neither *Fermi statistics* nor *Bose statistics* thus providing complex phases $e^{i\varphi}$, $\varphi \neq 0, \pi$ if two particles are exchanged or non-trivial transformations of the state vector if one particle moves around another. In fact, such particles can be observed in fractional quantum hall states [12] and are called *anyons* due to the occurrence of arbitrary phases.

Anyonic statistics occurs only in systems with two spacial dimensions, since the fundamental group of the punctured space with $D > 2$ is trivial. In other words: In three and more dimensions there is only *one* way of moving a particle around another up to homotopic deformations of the trajectory and consequently this trajectory is equivalent to the trivial one. It follows that the only possible particle statistics observable in high dimensional systems is the Fermionic and Bosonic.

In contrast, two-dimensional systems allow topologically non-trivial trajectories of a particle moving around another since the fundamental group of the punctured plane is non-trivial (it is isomorphic to $\mathbb{Z}$). As a consequence particle statistics in such systems is described by representations of the *braid group* B_2 [11]. If this representation is one-dimensional (thus abelian) the order of braiding anyons does not matter and they are called *abelian.* Representations of higher dimensions may cause non-commuting unitary transformations of the state by braiding two particles. Such particles are called *non-abelian anyons* [1, 11, 13].

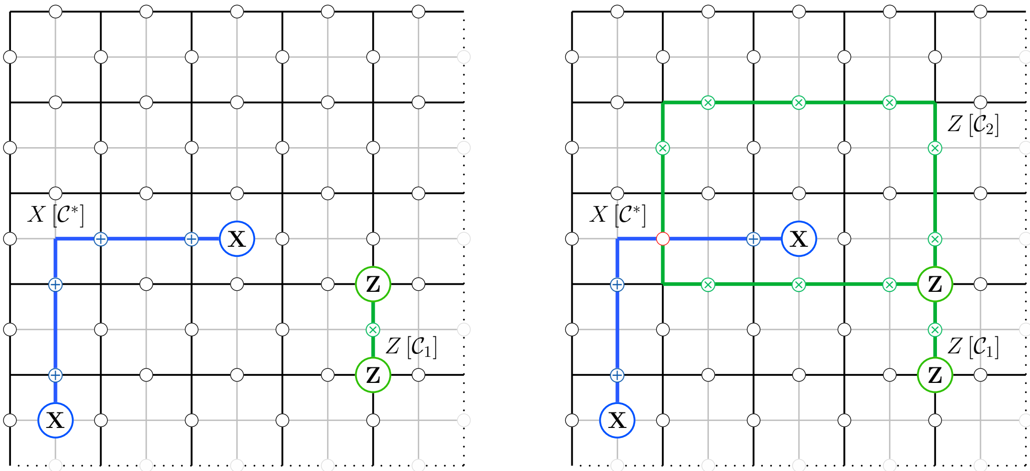


Figure 1.4 (Color online) : Two pairs of particles created by $Z[\mathcal{C}_1]$ and $X[\mathcal{C}^*]$ string operators (left picture). If the upper z-type particle is moved around the upper right x-type particle this is represented by the loop operator $Z[\mathcal{C}_2]$. Since $\mathcal{C}_2$ and $\mathcal{C}^*$ intersect at a single spin (or odd number in general) – marked by a red circle – this results in a minus sign (right picture).

Let us return to the Toric Code Model and its quasiparticles. Imagine that we created two pairs of them – one of each type – as illustrated in fig. 1.4 on the left. This can be done by applying string operators $Z[\mathcal{C}_1]$ and $X[\mathcal{C}^*]$ denoted by green and blue paths. To retrieve the *mutual* statistics between x- and y-type particles, move the upper z-type particle around the upper right x-type particle as illustrated in fig. 1.4 on the right. Mathematically this is represented by the loop operator $Z[\mathcal{C}_2]$. We find $X[\mathcal{C}^*]\,Z[\mathcal{C}_1]\,|\Psi\rangle$ for the initial state on the left and $Z[\mathcal{C}_2]\,X[\mathcal{C}^*]\,Z[\mathcal{C}_1]\,|\Psi\rangle$ for the final state on the right. Since $\{Z[\mathcal{C}_2], X[\mathcal{C}^*]\} = 0$ due to the red marked spin in the support of both operators, it follows $-X[\mathcal{C}^*]\,Z[\mathcal{C}_1]\,Z[\mathcal{C}_2]\,|\Psi\rangle = -X[\mathcal{C}^*]\,Z[\mathcal{C}_1]\,|\Psi\rangle$ for $\mathcal{C}_2$ is a *closed* path. Therefore a minus sign results from moving the particle of one type around a particle of another type and the elementary excitations of the TCM display *mutual abelian anyonic statistics.* Here *mutual* is an important information since both particle types obey Bose statistics if they are compared to particles of their own kind. This can be seen easily for there are only z- *or* x-type string operators involved.

We already used another property of the two elementary excitations[4] found in the TCM: The *fusion rules.* Obviously two particles of the same type annihilate each other if they occupy the same vertex or plaquette – they are their own antiparticles. These fusion rules are written as

$$e \otimes e = 1 \quad \textit{and} \quad m \otimes m = 1 \tag{1.49}$$

[4] In fact, there is a third one: The composition of x- and z-type particles yields y-type particles which are fermions.

where e (m) labels a z-type (x-type) particle and 1 denotes the trivial particle or *vacuum state*. Here e stands for *electric charge* and m for *magnetic vortex*. The reason for this naming scheme is the link between Toric Code Models and lattice gauge theories [1, 14].

1.3.3 Error correction

The reason for introducing the TCM was the notion of *fault tolerant quantum computation* [1] on a physical level. The basic idea proposed by Kitaev is the implementation of logical quantum gates acting on the logical space $\mathcal{PS}$ by creating, braiding and fusing anyons in a particular manner. This principle is known as *topological quantum computing*. Unfortunately it turns out that the TCM as presented here is too simple since it shows only abelian statistics. For a universal set of quantum gates to be implemented by braiding non-abelian anyons are required. Nevertheless one may flip and read out *logical* qubits stored in the TCM ground states ($|v_1, v_2\rangle$ is a logical 2-qubit state) by applying or measuring the X_i and Z_i loop operators. Although we cannot compute with these qubits in a quantum mechanical fashion, we look at the TCM as a storage for qubits. This storage turns out to implement *two* quantum error correction mechanisms on different layers.

To this end it is important to distinguish between *physical* and *logical* qubits. The physical qubits are represented by state vectors of the Hilbert spaces $\mathcal{H}_2$ or – as a composite system – by a state vector in $\mathcal{H}^N$ for N qubits. They are *physical* because these qubits would be implemented on a lattice in real space if one aims for a realization of the TCM (e.g. a realization with Rydberg atoms as physical qubits [15]). An implementation of interactions as described by H_{TCM} leads (disregarding any disturbance) to a ground state $|\Psi\rangle$ in the space $\mathcal{PS}$ formally protected by the stabilizer $\mathcal{S}$. This state space is also called *code space* since the ground states $|v_1, v_2\rangle$ describe the product of two 2-level systems, i.e. $\mathcal{PS} \cong \mathcal{H}_2 \otimes \mathcal{H}_2$. Therefore the ground states $|\Psi\rangle$ can be identified with the state of a 2-qubit system – these are the *logical* qubits. Note that the loop operators X_i and Z_i introduced above satisfy the same commutation relations as the Pauli matrices σ_i^x and σ_i^z. If we define $Y_i := iX_iZ_i$ we find

$$[A_i, B_i] = 2i\varepsilon_{A_iB_iC_i}C_i \quad \text{and} \quad \{A_i, B_i\} = 2\delta_{A_iB_i}\mathbb{1} \quad \text{where} \quad A_i, B_i, C_i \in \{X_i, Y_i, Z_i\} \tag{1.50}$$

which is the same algebra as for the Pauli matrices. Consequently the algebra generated by X_i and Z_i, $\mathcal{L}[\mathcal{PS}] := \langle X_1, X_2, Z_1, Z_2\rangle_{\mathbb{C}}$ with the composition of operators as multiplication acts on $\mathcal{PS}$ in the same way as the algebra of the Pauli matrices on a 2-qubit state space $\mathcal{H}^2$. This fact supports the notion of $\mathcal{PS}$ as the state space of *logical* qubits.

Storing few logical qubits in systems with much more physical qubits is a procedure not specific for the TCM but used extensively in the field of *quantum error correction*. This reduction of information density can be used (both in classical and quantum systems) to gain some error correction abilities, which leads to the first error correction layer present in Toric Code Models:

Algorithmic error correction Suppose an error occurs at spin $i \in E(\mathcal{L})$ on a TCM with $2N^2$ spins (i.e. on a $N \times N$ lattice), described mathematically by σ_i^x or σ_i^z on $\mathcal{H}^{2N^2}$. If $|\Psi\rangle \in \mathcal{PS}$ was the original state encoding 2 logical qubits, this operation yields $|\Psi\rangle_i^x := \sigma_i^x |\Psi\rangle$ for a x-type error at site i and $|\Psi\rangle_i^z := \sigma_i^z |\Psi\rangle$ for the corresponding z-type error. Obviously $|\Psi\rangle_i^{x,z} \notin \mathcal{PS}$ since there exist stabilizer operators A_s and B_p such that $A_s |\Psi\rangle_i^z = -|\Psi\rangle_i^z$ and/or $B_p |\Psi\rangle_i^x = -|\Psi\rangle_i^x$ iff $i \in s$ and/or $i \in p$. Note that

- one can detect and locate such an error by measuring all stabilizer generators A_s an B_p.
- the stored (coherent) quantum information is not lost, neither due to the error itself nor due to the following measurement of stabilizer operators.

First, errors described by unitary operators (here σ_i^x and σ_i^z) are invertible. This holds also for the stabilizer operators. Furthermore measuring the stabilizers does not change the corrupted state since $|\Psi\rangle_i^{x,z}$ is an eigenvector to each stabilizer. Consequently we can measure all stabilizer generators and apply σ_i^x or σ_i^z on the appropriate qubit according to the measurement outcome (which is called *error syndrome*) restoring the original quantum state $|\Psi\rangle$.

This leads to the question of how many errors may be *detected* and what condition must be fulfilled to *correct* such errors on a $N \times N$-spin lattice. To this end recall that the action of σ_i^x or σ_i^z can be described in terms of string operators $X\left[\mathcal{C}_{pq}^*\right]$ and $Z\left[\mathcal{C}_{pq}\right]$ (where single Pauli operators correspond to the simplest paths on $\mathcal{L}$ or $\mathcal{L}^*$, i.e. strings occupying only one edge). As we saw previously, the measurement of stabilizer generators yields information about the existence and location of the *endpoints* of a string and *not* its actual path. In the general case by measuring all stabilizer generators we obtain a somewhat "filtered" image of the errors occurred since the error syndrome contains information about the endpoint of error-chains only. This is illustrated in fig. 1.5.

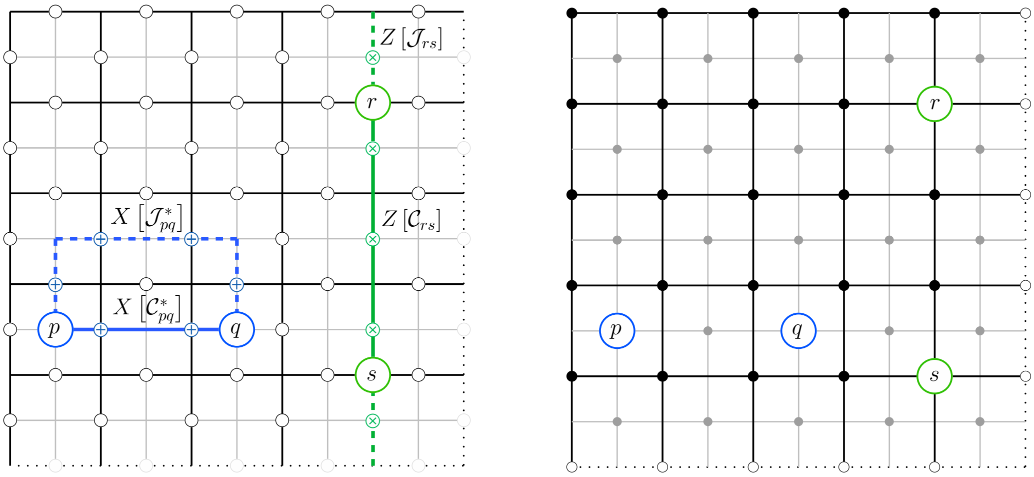

$Z\left[\mathcal{J}_{rs}\right]$
r
$X\left[\mathcal{J}_{pq}^*\right]$
$Z\left[\mathcal{C}_{rs}\right]$
$X\left[\mathcal{C}_{pq}^*\right]$
p
q
s
r
p
q
s

Figure 1.5 (Color online) : The left picture shows two error-chains and their string representation. Measuring the stabilizer generators yields the error syndrome depicted in the picture on the right. The left figure shows furthermore how string operators may be chosen to correct the errors according to the measured syndrome and the minimal distance assumption. Here the x-type error was corrected whereas the "correction" of the z-type error destroyed one logical qubit by flipping it.

If one tries to correct these errors *only using the information contained in the error syndrome*, a problem occurs if the error-chain has length $l > \lfloor \frac{N-1}{2} \rfloor$. At first glance this problem seems to originate from the implementation of the error correction algorithm itself. But as one thinks about this carefully, it becomes clear that there is a deeper connection to information theory and the behaviour of information as (physical?) entity. Now, what is this problem alike?

The picture on the right of fig. 1.5 shows how errors can be corrected by applying string operators $X\left[\mathcal{J}_{pq}^*\right]$ and $Z\left[\mathcal{J}_{rs}\right]$ to the corrupted state in question. By applying string operators with strings attached to the sites where the error syndrome indicates endpoints of strings *with the same type*, one creates loops of z- and x-type, respectively. These act trivially on $\mathcal{PS}$ and the original states are recovered. The essential part here is that the *physical* error graven into a state due to some earlier transformation does not depend on the actual path of the string operator describing it. Analogously the consequence of a *physical* error correction does not depend on the actual path of the string operator used. Although different strings lead to different physical actions (the error itself and the subsequent correction are physical events) the *state* transforms equally if two strings are homologous. There is no way *measuring* the actual path afterwards since this information is not contained in the state itself. To say, TCM ground states have the property of "factoring out" physical actions with respect to some equivalence relation which is connected to the first homology group of the considered surface (see sec. 1.4.1).

The problem arising is due to the freedom choosing the path connecting two sites p and q where errors have been detected. Whereas two homologous paths (on the 1-torus one can just think of homotopic paths, i.e. paths with fixed endpoints which can be deformed continuously into each other) have the same effect, two non-homologous paths lead to non-trivial actions on the original state, thus destroying the stored information rather than recovering it.

Formally this reads (see fig. 1.5)

$$X\left[\mathcal{J}^*_{pq}\right]X\left[\mathcal{C}^*_{pq}\right]=X\left[\mathcal{J}^*_{pq}+\mathcal{C}^*_{pq}\right]=\mathbb{1}\quad\text{but}\quad Z\left[\mathcal{J}_{rs}\right]Z\left[\mathcal{C}_{rs}\right]=Z\left[\mathcal{J}_{rs}+\mathcal{C}_{rs}\right]=Z_1 \tag{1.51}$$

since $\left[\mathcal{J}^*_{pq}+\mathcal{C}^*_{pq}\right]=0$ and $\left[\mathcal{J}_{rs}+\mathcal{C}_{rs}\right]\neq 0$. Here $[\cdot]$ denotes the canonical surjection into the first homology group of the lattice. Note that $X\left[\mathcal{J}^*_{pq}+\mathcal{C}^*_{pq}\right]$ acts as the identity only on $\mathcal{PS}$.

For an error correction operator to succeed $\left[\mathcal{J}_{rs}+\mathcal{C}_{rs}\right]=0$ is a necessary and sufficient condition. The problem is that we cannot be sure in general if we chose the correction path appropriately. It may happen that it forms a non-trivial loop and consequently flips the corresponding logical qubit instead of recovering the old one. This problem is solved partially using a rule by which the path connecting given endpoints r and s or p and q is chosen. To this end introduce a distance measure[5] on $V(\mathcal{L})$ and $P(\mathcal{L})$ by

$$d(r,s) \quad := \quad \min_{\mathcal{C}\,:\,r,s\in\mathcal{C}} |\mathcal{C}| \qquad r,s\in V(\mathcal{L}) \tag{1.52a}$$

$$d(p,q) \quad := \quad \min_{\mathcal{C}^*\,:\,p,q\in\mathcal{C}^*} |\mathcal{C}^*| \qquad p,q\in P(\mathcal{L}) \tag{1.52b}$$

where $|\cdot|$ denotes the length of the path (i.e. the number of its edges) and the min runs over all paths containing vertices r and s or faces p and q, respectively. Suppose the error syndrome detects errors at vertices r and s (faces p and q). To correct this error the following paths are chosen (up to homology)

$$\mathcal{J}_{rs} \quad : \quad |\mathcal{J}_{rs}|=d(r,s) \tag{1.53a}$$

$$\mathcal{J}^*_{pq} \quad : \quad \left|\mathcal{J}^*_{pq}\right|=d(p,q) \tag{1.53b}$$

If there are homologously different paths with the same minimal length one of them is chosen arbitrarily. *Assuming* that a corrupted state was affected by at most $k=\lfloor\frac{N-1}{2}\rfloor$ errors the rule stated above leads in any case to a successful recovery of the original state. If there are more than k errors, this holds not true in any case and therefore we cannot be sure that the qubits obtained after the application of a string operator are the same as before. This statement equals the conclusion that more than k errors cannot be corrected, even if in some cases the new state resembles the original one.

Physical error correction An interesting property of the TCM is that it is *not only* a code featuring quantum error correction procedures as explained above but provides robustness against noise on a *physical level* [1]. This is due to the Hamiltonian

$$H_{TCM}=-\sum_{s\in V(\mathcal{L})}A_s-\sum_{p\in P(\mathcal{L})}B_p$$

which introduces an *energy penalty* for each pair of anyons (i.e. errors described by σ^x and σ^z). Without thermal fluctuations ($T=0$) Toric Code systems give rise to a *gapped topological phase*, meaning that elementary excitations come along with a finite energy cost and the ground state degeneracy, used to encode quantum information coherently, cannot be lifted by means of local[6] perturbations. Thus errors (i.e. operations transforming one ground state into another) can only disturb or destroy coherently stored information if they affect topologically non-trivial parts of the system.

In the case of Toric Code Models an error that destroys the stored information at least affects physical qubits along a homologously non-trivial path. In other words: One must create a pair of anyons, pull one of them along a non-trivial path around the torus and annihilate it with

[5] This notion of "distance" is also known as *geodesic distance*.

[6] In this context "local perturbations" or "local operators" denote operators with support that scales *not* with the system size. E.g. operations acting on two spins are *local* whereas non-trivial loop operators are *nonlocal*. However, in the field of entanglement theory operations on two spins may be denoted *nonlocal* if they increase entanglement measures whereas even non-trivial loop operators are *local* unitaries.

its antiparticle. Hence no local perturbation affects the coherently stored information since any *confined* pair of anyons maps the protected space onto an orthonormal state space. This mapping is undone by annihilating both anyons if their trajectory is trivial with respect to homology.

Since a phase at $T = 0$ is not viable in real physical systems the question arises whether some properties characterizing the topological phase survive at finite temperatures. To this end several approaches were taken to detect and characterize topological phases [16–18] which led (amongst others) to a quantity called *topological entropy* measuring topologically non-trivial long range entanglement [16, 17, 19]. It was shown that the topological entropy of Toric Code Models in two dimensions is fragile with respect to thermal fluctuations due to their disruptive effect on the critical loop structure responsible for the topological order [20]. However, it was argued that topological phases may survive at finite temperatures if the Toric Code Model is extended to three dimensions [21]. In two dimensions it was shown[7] that the expectation values of the loop operators vanish identically for finite temperatures, i.e. $\langle Z_1\rangle = \langle Z_2\rangle = \langle X_1\rangle = \langle X_2\rangle = 0$ if $T > 0$, for both finite systems and system in the thermodynamic limit [18,22]. This is caused by fluctuating strings between thermal induced pairs of anyons leading to time dependent eigenvalues of non-trivial loop operators that may also depend on their actual shape.

1.4 Basics of graph theory and algebraic topology

1.4.1 Algebraic topology

The following section will *not* give a formal introduction of algebraic topology since this goes beyond the scope of this thesis. For proofs and basic definitions the reader is referred to textbooks of (algebraic) topology, e.g. [23].

The main purpose of this section is to provide a link between notions of algebraic topology and the structure featured by Toric Code Models. Hence we will focus on closed paths (cycles) on compact 2-manifolds.

A first approach Since important properties of Toric Code Models (e.g. the ground state degeneracy) are influenced by the number of holes in the surface the system lives on, a topological entity is needed which measures such structural properties. One of the most simple mathematical objects measuring the number of holes is the *fundamental group* $\pi_1(X, x_0)$ defined for a punctured topological space X. It is obtained as quotient of the group $\{\mathcal{C} : [0, 1] \to X \,|\, \mathcal{C}(0) = x_0 = \mathcal{C}(1)\}$ of closed paths $\mathcal{C}$ running through x_0 (with concatenation of paths as multiplication) with respect to homotopy. Thus two paths lie within the same equivalence class in $\pi_1(X, x_0)$ iff there is a continuous deformation transforming one path into the other. For instance the blue and yellow loops drawn in fig. 1.6 on the 2-torus are homotopic whereas the green and red paths are not for the hole separating them.

Let us check whether the notion of *homotopic* paths reflects the notion of *topologically equivalent* paths known from the Toric Code Model correctly. To this end let $\mathcal{L}$ be a (sufficient fine) lattice embedded into the surface depicted in fig. 1.6. We called two closed paths $\mathcal{C}_1$ and $\mathcal{C}_2$ on $\mathcal{L}$ topologically equivalent if their corresponding loop operators act equally on the protected states, i.e. $Z[\mathcal{C}_1] = Z[\mathcal{C}_2]$ on $\mathcal{PS}$ (the same holds for loops on the dual lattice $\mathcal{L}^*$ and x-type loop operators).

The question is whether $Z[\mathcal{C}_1] = Z[\mathcal{C}_2]$ is equivalent to $\mathcal{C}_1$ and $\mathcal{C}_2$ being *homotopic*. In fact, for the yellow and green paths this is certainly true. They are homotopic and their corresponding z-type loop operators return the same topological quantum number (or their x-type loop operators flip the same logical qubit). This can be seen easily since $Z[\mathcal{C}_{yellow}]$ equals $Z[\mathcal{C}_{blue}]$ up to some plaquette operators B_p (or star operators A_s) filling the surface of the "tube" between them.

[7] By mapping the Toric Code Model onto two decoupled Ising chains.

However, our approach fails due to the example given by the green and red loops. Despite the lack of homotopy it holds $Z[\mathcal{C}_{green}] = Z[\mathcal{C}_{red}]$ on $\mathcal{PS}$. This follows since the area enclosed by $\mathcal{C}_{red}$ and $\mathcal{C}_{green}$ may be filled with plaquette or star operators despite the "hole" piercing the surface. Obviously it is *not* important if $\mathcal{C}_1$ can be deformed into $\mathcal{C}_2$ continuously but rather if the combination of both forms a *boundary of an area* embedded into the considered surface. This relation leads to the notion of *homologous* paths and the *first homology group*.

Simplicial homology The following introduction of the first homology group H_1K of a simplicial complex K is purely mathematical and is commonly used for introductory purposes. However, if one thinks of a generalised Toric Code Model (sec. 1.5) defined on a triangular lattice $\mathcal{L}$ together with its z-type loop operators and plaquette operators B_p the analogue is quite demonstrative.

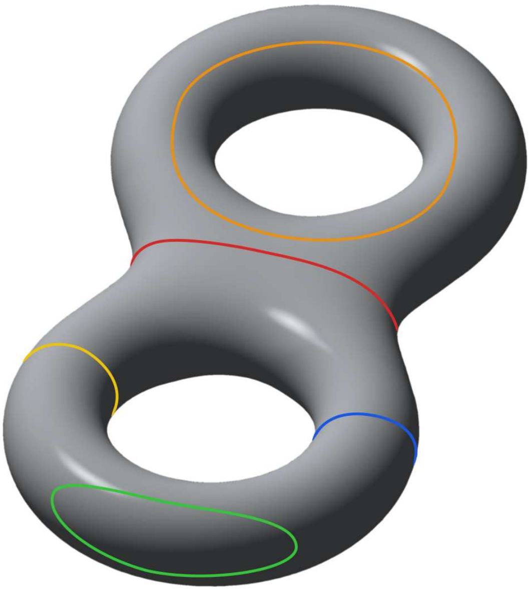


Figure 1.6 (Color online) : A compact 2-manifold without boundary and genus 2. It is called 2-*torus* since it is the connected sum[8] of two tori. The relations between the closed paths drawn on the surface are described in the text.

In topology a *n-simplex* is the generalization of a triangle (0-simplex: point, 1-simplex: line segment, 2-simplex: triangle, 3-simplex: tetrahedron etc.). A simplicial complex K is a set of such simplices that satisfies further conditions. E.g. the triangular lattice $\mathcal{L}$ can be seen as a simplicial complex made of 2-simplices (a combination of triangles connected by common edges only). Since we are not interested in the common case let $K = \mathcal{L}$ from now on. $s = [v_0, v_1, v_2]$ denotes the 2-simplex (i.e. a plaquette) spanned by its three vertices $v_0, v_1, v_2 \in V(\mathcal{L})$ whereas $e = [v_0, v_1]$ denotes the 1-simplex (i.e. an edge) spanned by its two adjacent vertices v_0, v_1. 0-simplices $v = [v]$ are the vertices of $\mathcal{L}$. Now define a *p-chain* as the formal sum $c = \sum_i \lambda_i x_i$ where $\lambda_i \in \mathbb{F}_2$ and x_i denote p-simplices of $\mathcal{L}$. These sums may be seen as arbitrary collections of faces/edges/vertices of $\mathcal{L}$. Clearly the set C_p of all possible (finite) sums c becomes an abelian group if we define

$$c + c' := \sum_i (\lambda_i + \lambda_i')x_i \tag{1.54}$$

$(C_p, +)$ is called the *group of p-chains*. Now define a group homomorphism $\partial_p : C_p \to C_{p-1}$, called the *boundary map*, by its action on p-simplices:

$$\partial_p s := \sum_{j=0}^{p} [v_0, \dots, \hat{v}_j, \dots, v_p] \quad \text{where} \quad [v_0, \dots, \hat{v}_j, \dots, v_p] = [v_0, \dots, v_{j-1}, v_{j+1}, \dots, v_p] \tag{1.55}$$

For arbitrary p-chains we have $\partial_p c = \sum_i \lambda_i \partial_p x_i$. In our case the boundary of a 2-simplex (triangle) is given by the formal sum of its three bounding edges as 1-simplices. Furthermore the boundary of an 1-simplex (edge) is simply the formal sum of its adjacent vertices.

A p-chain c is called *p-cycle* iff $\partial_p c = 0$. In fact, 1-cycles are just the (combinations of) closed paths $\mathcal{C}$ on the lattice $\mathcal{L}$. This is easy to verify since the relation $\partial_1 c = 0$ tells us that c is a collection of edges such that each vertex is adjacent to an even number of these edges.

A p-chain c is called *p-boundary* iff there exists a $p+1$-chain c' such that $c = \partial_{p+1} c'$. In our case the 1-boundaries are exactly the boundaries of arbitrary collections of triangles. At this point

[8] Formally one writes $\mathbb{T}\#\mathbb{T}$ where $\mathbb{T}$ denotes a torus.

it becomes clear that the theory developed above is deeply connected to loop operators $Z[\mathcal{C}]$ of Toric Code Models.

Let $\mathcal{Z}_p$ and $\mathcal{B}_p$ denote the sets of p-cycles and p-boundaries. Obviously $\mathcal{Z}_p = \ker \partial_p$ and $\mathcal{B}_p = \operatorname{im} \partial_{p+1}$ by definition. Since ∂ is a homomorphism of (abelian) groups it follows immediately that $\mathcal{Z}_p$ and $\mathcal{B}_p$ are (normal) subgroups of C_p. It can be shown by straightforward calculation that every p-boundary is a p-cycle (since $\partial_p \partial_{p+1} c' = 0$). Hence we find $\mathcal{B}_p \triangleleft \mathcal{Z}_p$ ($\mathcal{B}_p$ is a normal subgroup of $\mathcal{Z}_p$).

The *pth homology group* $H_p \equiv H_p K$ of the simplicial complex K (here a triangular lattice $\mathcal{L}$) is defined as the factor group of p-cycles with respect to p-boundaries

$$H_p := \mathcal{Z}_p \Big/ \mathcal{B}_p \tag{1.56}$$

Note that H_p is abelian, which is not true in general for the fundamental group π_1. We call two p-cycles c_1 and c_2 *homologous* ($c_1 \sim c_2$) iff they belong to the same homology class $[c_1] = [c_2] \in H_p$. Consider the special case $p = 1$ for our triangular lattice $\mathcal{L}$ on the 2-torus depicted in fig. 1.6. Here $\mathcal{L}$ is called the *triangulation* of the surface. Two 1-cycles (i.e. loops) are homologous if they differ only by the boundary of an arbitrary shaped (not necessarily connected) area embedded into the surface. Thus the yellow and blue loops belong to the same homology class whereas the orange loop is not homologous to any other drawn loop. Furthermore the green and red paths are homologous and obviously $[\mathcal{C}_{green}] = [\mathcal{C}_{red}] = 0 \in H_1(\mathbb{T}\#\mathbb{T})$ since both are 1-boundaries. This is equivalent to say that $Z[\mathcal{C}_{green}] = Z[\mathcal{C}_{red}] = \mathbb{1}$ on $\mathcal{PS}$ in the Toric Code Model. More generally: Two z-type loop operators $Z[\mathcal{C}_1]$, $Z[\mathcal{C}_2]$ are equivalent on $\mathcal{PS}$ iff $[\mathcal{C}_1] = [\mathcal{C}_2]$ with respect to the first homology group of the lattice $\mathcal{L}$ since this is the case only if the loop operators are equal up to some plaquette operators B_p. Thus the algebra of z-type (x-type) loop operators is determined by the first homology group of the lattice $\mathcal{L}$ (dual lattice $\mathcal{L}^*$)[9].

Homology and ground state degeneracy In subsec. 1.4.2 we will show that the ground state degeneracy of a generalised Toric Code Model embedded into a surface of genus g is $\dim \mathcal{PS} = 4^g$ by means of graph theory. However, there is another possibility to derive the degeneracy in terms of algebraic topology.

We saw that each logical qubit encoded in the state of a Toric Code system is represented by a topological quantum number v_i. For each quantum number v_i there is a z-type loop operator Z_i such that $Z_i |v_1, \ldots, v_n\rangle = v_i |v_1, \ldots, v_n\rangle$. Consequently n independent logical qubits correspond to n independent[10] loop operators Z_i. As mentioned above there is a one-to-one correspondence between loop operators and homology-classes in H_1. It is a well known fact from algebraic topology that $H_1(\Sigma_g) = \mathbb{F}_2^{2g}$ where Σ_g denotes a compact orientable surface of genus g [11] and it follows $\operatorname{rank} H_1(\Sigma_g) = 2g$. Thus there are $2g$ independent homology classes for a surface of genus g if the cycle group is defined as above. Consequently there are $2g$ logical qubits encoded in the Toric Code. This yields 2^{2g} orthonormal ground states. Hence we derived $\dim \mathcal{PS} = 4^g$.

1.4.2 Graph theory

Graph theory as a field of discrete mathematics deals with fundamental objects called *graphs* and their properties. The theory of such objects turns out to be of great importance for a variety of sciences such as theoretical informatics, complexity theory and theoretical physics. There are several different notions of a graph – two of them are used for the description of special spin systems in subsequent sections. Throughout this thesis in the majority of cases we adopt the notation of [24].

[9] This statement holds true even for non-triangular lattices.

[10] Here “independent” is meant in terms of the abelian group generated by z-type loop operators.

[11] Usually homology groups are computed over $\mathbb{Z}$. Then it holds $H_1(\Sigma_g) = \mathbb{Z}_2^{2g}$.

Definition 1.4.1: (Simple graph, Multi graph). *Let V be an arbitrary set.*

(i) Let $E \subseteq [V]^2$ be a subset consisting of 2-element sets of V. Elements of V are called ***vertices****, elements of E* ***edges****. The pair G=(V,E) is called an* ***undirected simple graph*** *or simply* ***graph****. By $V(G)$ we denote the set of vertices whereas $E(G)$ denotes the set of edges. Write $|G| := |V(G)|$ for the* ***order*** *of the graph G (i.e. the number of vertices) and $\|G\| := |E(G)|$ for the number of edges. If $|G| < \infty$ G is called a* ***finite*** *graph. If nothing is stated about $|G|$ the graph is assumed to be finite.*

(ii) In contrast to (i) let $E \subseteq [V]^2$ be a ***multiset*** *consisting of 2-element sets of V. Then the tuple G=(V,E) is called an* ***undirected multigraph*** *or simply* ***multigraph****.*

A multiset M is a generalization of a set S. The latter allows elements to occur at most once, i.e. $S = \{a,b,c\} = \{a,a,b,c,c,c\}$ whereas a multiset allows more than one element of each kind, i.e. $M = \{a,a,b,c,c,c\} \neq \{a,b,c\}$ (note that this is not a formal definition of a multiset). The number of occurrences of an element is called the *multiplicity*. Applied to the definition of graphs given above this leads to different ways two vertices p and q can be connected. By *connected* we mean that there exists (at least) one edge $e \in E(G)$ such that $p, q \in e$. The definition of a simple graph ensures that there is at most *one* edge connecting two vertices whereas the definition of a multigraph allows *multiple* edges connecting two vertices. Fig. 1.7 illustrates these two graph types.

Remark 1.4.2. *Every simple graph is also a multigraph. We denote only true multigraphs as such, i.e. graphs that are multigraphs but no simple graphs.*

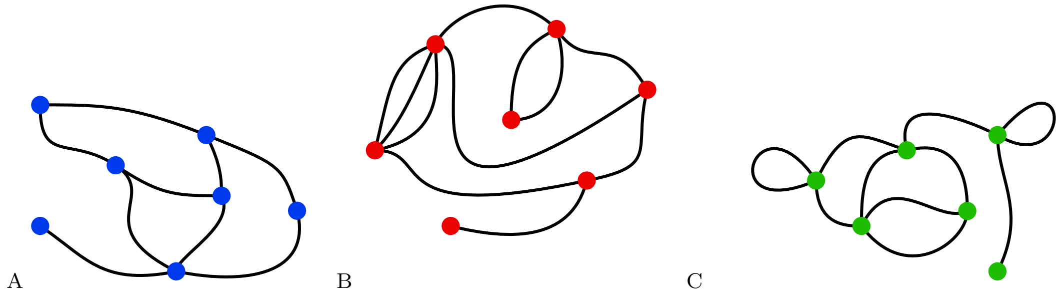


Figure 1.7 (Color online) : Examples of different graph types. The blue graph in A is simple in the sense of def. 1.4.1 whereas the red one in B is a true multigraph due to the multiple edges. The green one in C is not defined in 1.4.1 and is sometimes referred to as *pseudograph*. We are not going to deal with graphs of the green type (i.e. graphs with cycles of length 1). All these graphs are finite, undirected and connected.

Two graphs G_1 and G_2 may be related to each other. The most important (equivalence) relation between graphs is called *graph isomorphism*.

Definition 1.4.3: (Graph isomorphism). *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two (simple) graphs. Then we call them* ***isomorphic*** *($G_1 \simeq G_2$) if there exists a bijection $\pi : V_1 \to V_2$ such that*

$$\{p,q\} \in E_1 \quad \Leftrightarrow \quad \{\pi(p), \pi(q)\} \in E_2 \tag{1.57}$$

Obviously $\simeq$ is an equivalence relation. Consequently the class of all graphs decomposes into equivalence classes of pairwise isomorphic graphs. Such classes may be drawn as *unlabeled* graphs, graphs without numbers associated to the vertices, that is. Therefore graph isomorphisms give rise to a more abstract view towards graphs by extracting their pure structure. The equivalence classes under graph isomorphisms are called *abstract graphs*.

Definition 1.4.4: (Subgraph, Tree, Leaf, Spanning tree). *Let $G = (V, E)$ and $H(W, F)$ be (simple or multi) graphs.*

*(i) H is called a **subgraph** of G ($H \subseteq G$) iff $W \subseteq V$ and $F \subseteq E$.*

*(ii) G is called a **tree** iff it is simple, connected and has no cycles.*

*(iii) A vertex $v \in V$ is called **leaf** iff it has only one adjacent edge $e = \{v, v'\} \in E$. Sometimes we will call e leaf and v (v') its **outer (inner) vertex**.*

*(iv) H is called a **spanning tree** of G iff H is a tree, $V = W$ and $F \subseteq E$.*

According to def. 1.4.4 a spanning tree of a graph G is a subgraph of the latter. Spanning trees of a given graph may be constructed by deleting a set $E' \subseteq E$ of edges such that

1. there is no cycle $\mathcal{C}$ in G where $\mathcal{C}$ contains only edges in $E \setminus E'$.

2. E' is minimal, i.e. if one removes an arbitrary edge from E' the first condition is violated.

It is easy to see that spanning trees are not unique (even under graph isomorphisms). See fig. 1.8 for an illustration of the procedure explained above.

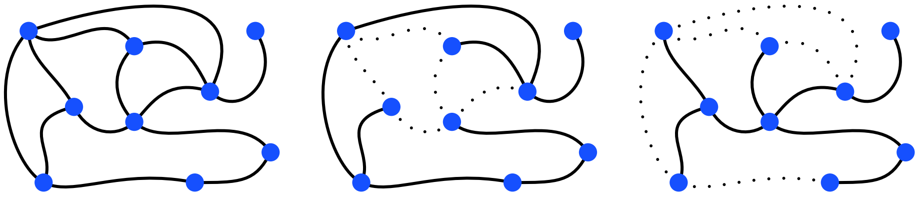


Figure 1.8 (Color online) : The graph on the left is an arbitrary simple graph which is not a tree since there are several cycles. By deletion of edges supporting a cycle (illustrated in the remaining two graphs by dashed edges) a spanning tree of the graph is generated. Such spanning trees are not unique (even under graph isomorphisms) as the two different spanning trees indicate.

There are several ways of transforming a graph locally or globally. To this end let us introduce the notation $G - a = (V, E) - a := (V \setminus \{a\}, E \setminus \{e \in E \,|\, a \in e\})$ to describe the deletion of a vertex $a \in V$ and its adjacent edges. Analogously we define the deletion of an edge $e \in E$ via $G - e = (V, E) - e := (V, E \setminus \{e\})$. These shortcuts may be generalized straightforwardly to the subtraction of sets of vertices and edges. Furthermore one can define an additive structure on the set of all graphs with the same vertex set V. This structure is given by modulo 2 addition of edges, i.e. two graphs $G = (V, E)$ and $H = (V, F)$ can be merged by $G + H := (V, E \triangle F)$ where $\triangle$ denotes the symmetric difference of sets. Furthermore one defines the union $G_1 \cup G_2 := (V_1 \cup V_2, E_1 \cup E_2)$ and the intersection $G_1 \cap G_2 := (V_1 \cap V_2, E_1 \cap E_2)$ of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$.

Definition 1.4.5: (Neighbourhood, Complete graph, Induced subgraph, Complementation). *Let $G = (V, E)$ be a (simple) graph, $a \in V$ and $U \subseteq V$.*

*(i) $N_a := \{p \in V \,|\, \{p, a\} \in E\} \subseteq V$ is called the **neighbourhood** of vertex a.*

*(ii) The **complete graph** of an arbitrary vertex set U is denoted by $\langle U \rangle := \big(U, [U]^2\big)$.*

*(iii) $G[U] := (U, \{e = \{p, q\} \in E \,|\, p, q \in U\}) \subseteq G$ is called the **induced subgraph** over U in G.*

*(iv) $\overline{G} := \big(V, [V]^2 \setminus E\big) = G + \langle V \rangle$ is called the **complement** of G.*

*(v) $\tau_a(G) := G + \langle N_a \rangle$ is called the **local complement** of G at vertex a.*

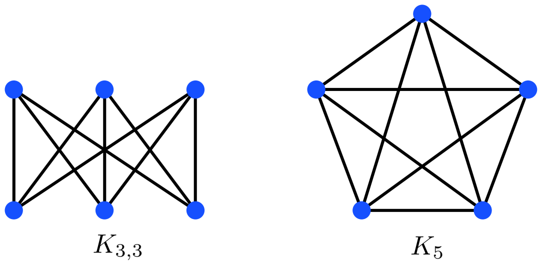


Figure 1.9 (Color online) : The two non-planar graphs used in Kuratowski's theorem. $K_{3,3}$ and K_5 denote the complete bipartite graph on six vertices and the complete graph on five vertices, respectively.

Particularly the *local complementation* of graphs will be important later on (see sec. 1.6). Local complementation induces a second equivalence relation on the class of graphs with the same vertex set V. According to this relation two graphs G_1 and G_2 are equivalent ($G_1 \sim_{\text{LC}} G_2$) iff there exists a finite sequence of local complementations τ_{a_i}, $(a_i)_{i=1}^N \subseteq V$ such that $\prod_{i=1}^N \tau_{a_i} G_1 = G_2$. The distinction of different classes with respect to local complementation was investigated in [25, 26].

The following definition allows one to speak of graphs with the same vertex set as *vectors*:

Definition 1.4.6: (Edge space). *Let V be a set of vertices. Then $\mathcal{E}(V) = \mathcal{P}\left([V]^2\right)$ (here $\mathcal{P}$ denotes the power set) becomes a vector space over the Galois field $\mathbb{F}_2$ if one defines for all $E, F \in \mathcal{E}(V)$*

$$E + F := E \triangle F \qquad \textit{and} \qquad 0 \cdot E := \emptyset \qquad \textit{as well as} \qquad 1 \cdot E := E$$

*where $\triangle$ denotes the symmetric difference. Hence $\mathcal{E}(V) = \operatorname{span}_{\mathbb{F}_2}[V]^2$ is the free vector space generated by the edges of the complete graph $\langle V \rangle$. It is called **edge space** over V.*

Two graphs $G = (V, E)$ and $H = (V, F)$ may be identified with their edge sets as vectors $E, F \in \mathcal{E}(V)$, for the sake of simplicity we write $G, H \in \mathcal{E}(V)$. Consequently the addition of graphs defined previously translates into the addition of edge-vectors: $G + H = (V, E + F) = (V, E \triangle F)$. From this point of view a local complementation τ_a, $a \in V$ is a mapping $\tau_a : \mathcal{E}(V) \to \mathcal{E}(V)$ from the edge space into itself.

Let us now turn towards *graph embeddings*. A graph embedding is a representation of a graph as a subset of a compact, connected 2-manifold. We are not going to define a graph embedding formally since this is a rather technical issue and is not necessary for the physics described later on. Just think of a graph embedding as a drawing of a graph on a 2-manifold (e.g. a torus) without edges intersecting each other. Graphs that are embeddable into the plane are called *planar graphs*. An important theorem providing information about the planarity of a given graph is the following:

Theorem 1.4.7: (Kuratowski's theorem). *Let $G = (V, E)$ be an arbitrary finite graph. Then G is planar iff it does not contain a subgraph $G' \subseteq G$ that is a subdivision of $K_{3,3}$ or K_5.*

For a proof see ref. [24]. The two graphs mentioned by Theorem 1.4.7 are depicted in fig. 1.9. $K_{3,3}$ denotes the complete bipartite graph on 6 vertices whereas K_5 denotes the complete graph on 5 vertices. A *subdivision* of a given graph G can be obtained by cutting an edge and gluing both ends together by inserting a new vertex at the cut.

One may ask, if every possible (finite simple or multi) graph can be embedded on an appropriate surface. In fact, it is possible as one can easily deduce: Start with the sphere S^2 and draw the graph on its surface. If the graph is non-planar this is not possible without intersecting edges[12]. Now add a handle to the sphere (hence increasing the genus of the surface) for every intersection of edges that is not removable on the sphere. These handles act as "bridges" for the intersecting edges since one of them can now be drawn above the other without intersecting it.

This construction leads to the following

[12] A graph is embeddable on the sphere S^2 if and only if it is planar. To see this, just project the sphere onto the plane by means of a stereographic projection.

Definition 1.4.8: (Genus of a graph). *Let $G = (V, E)$ be a (simple or multi) graph. Then the* ***(oriented) genus*** *$g \equiv g[G]$ of the graph is defined as the minimal genus of all (oriented) surfaces Σ such that G is embeddable on Σ.*

That the genus of a graph is well defined (i.e. always exists) follows from the considerations above that there is always an orientable surface a given graph is embeddable into.

The following notions of *dual graphs* and *faces* are only defined for embedded graphs and are important for Toric Code Models and surface codes in general (see sec. 1.5):

Definition 1.4.9: (Faces, Dual graph). *Let $G = (V, E)$ be a (simple or multi) graph embedded on a surface Σ.*

(i) The set of ***faces*** *defined by this embedding is denoted by $P_\Sigma(G)$ or simply $P(G)$ if there is no danger of confusion. A face or plaquette $p \in P_\Sigma(G)$ is identified with the set of edges on its boundary. Thus we write $e \in p$ for these edges.*

(ii) The ***dual graph*** *$G^* = (V^*, E^*)$ is defined as follows: Set $V^* := P_\Sigma(G)$. Two dual vertices $v_1^*, v_2^* \in V^*$ are connected iff there is a common edge in G to v_1^* and v_2^* as faces. Formally this reads*

$$\forall\, v_1^*, v_2^* \in V^* \,:\, \{v_1^*, v_2^*\} \in E^* \Leftrightarrow \exists\, e \in E \,:\, (e \in v_1^* \wedge e \in v_2^*) \tag{1.58}$$

Thus the vertices/edges/faces of G^* correspond to the faces/edges/vertices of G.

In the case of Kitaev's Toric Code G is substituted by $\mathcal{L}$ and called a *lattice* due to its spacial periodicity whereas G^* is known as $\mathcal{L}^*$ and called the *dual lattice*.

In conclusion we mention an important relation between vertices, edges and faces of a graph on one side and its genus on the other side:

Theorem 1.4.10: (Euler's formula). *Let $G = (V, E)$ be a (simple or multi) graph of genus g embedded on a surface Σ of genus $g\,[\Sigma] = g$. Then it holds*

$$v + f - e = \chi = 2 - 2g \tag{1.59}$$

where v, e and f denote the number of vertices, edges and faces, respectively. χ is called the ***Euler characteristic*** *of G.*

The TCM and Euler's formula Let us combine three theories we introduced so far: The Toric Code Model, the stabilizer formalism and graph theory. As we will show Euler's formula gives rise to a simple proof of the degeneracy (or dimension) of a Toric Code ground state space on arbitrary compact, orientable 2-manifolds. In section 1.3 we saw that the ground state space $\mathcal{PS}$ of a Toric Code stabilizer $\mathcal{S}$ with lattice $\mathcal{L}$ embedded on the torus is fourfold degenerate. Now let $\mathcal{L}$ be a sufficient fine[13] lattice embedded into a surface of genus g. We may take the lattice $\mathcal{L}$ as a graph embedded into the surface. However, in terms of a TCM vertices correspond to A_s operators and faces can be identified with B_p operators. Not least every edge represents a spin in our physical model. Therefore Euler's formula reads

$$|V\left(\mathcal{L}\right)| + |P\left(\mathcal{L}\right)| - N = 2 - 2g \tag{1.60}$$

if we consider a system of N spins. According to Proposition 1.2.2 it holds for the degeneracy $\dim \mathcal{PS} = 2^{N-d}$ where d denotes the rank of the stabilizer, i.e. the number of independent generators. Since $\prod_s A_s = \mathbb{1} = \prod_p B_p$ it follows $d = |P\left(\mathcal{L}\right)| + |V\left(\mathcal{L}\right)| - 2 = N - 2g$. Finally

$$\dim \mathcal{PS} = 2^{N-(N-2g)} = 2^{2g} = 4^g \tag{1.61}$$

This shows that the degeneracy of a Toric Code system described by the Hamiltonian H_{TCM} only depends on the topology of the surface and each "hole" leads to a pair of qubits that can be stored in the ground state configuration. Furthermore the result obtained in subsec. 1.4.1 by means of algebraic topology was verified.

[13] By "sufficient fine" we mean that $\mathcal{L}$ is not embeddable into a surface of lower genus.

1.5 Generalised Toric Code Models: Surface codes

Usually the Toric Code is defined on a square lattice embedded into a 1-torus. This is not necessary since the topological properties (see sec. 1.4.1) are neither restrained to square lattices nor to surfaces of genus 1. Hence one defines *generalised Toric Code Models* or *surface codes* [1, 27, 28] as follows:

Definition 1.5.1: (Surface codes). *Let G be a (multi) graph of genus g embedded into a surface Σ of genus $g\,[\Sigma] = g$. $V(G)$ denotes the set of vertices and $P_\Sigma(G)$ the set of faces defined by the embedding on Σ. Define (generalised) star and plaquette operators*

$$A_s := \bigotimes_{i\in s} \sigma_i^x \qquad \textit{and} \qquad B_p := \bigotimes_{i\in p} \sigma_i^z \tag{1.62}$$

*for each vertex $s \in V(G)$ and face $p \in P_\Sigma(G)$. Then $\mathcal{S}[G] := \operatorname{span}\{A_s, B_p\}_{s\in V(G), p\in P_\Sigma(G)}$ is the stabilizer of the **surface code** corresponding to G and Σ. Its degeneracy is $\dim \mathcal{PS} = 4^g$ in the case of closed boundary conditions*[14] *(see proof in subsec. 1.4.2).*

All properties derived for the classic Toric Code Model in sec. 1.3 hold for surface codes as well. Hence the algebra of operators on the code space (the state space of logical qubits) is given by z- and x-type loop operators Z_i and X_i for $1 \le i \le 2g$ with support on the graph G or its dual G^*, respectively. The loops defining Z_i and X_i operators are called non-trivial *cycles* and *cocycles*. In the following (if not stated otherwise) we will use Toric Code systems in this broader sense and call them simply *(generalised) Toric Code Models*.

Open boundary conditions Up to now the dimension of the protected subspace $\mathcal{PS}$ was determined solely by the genus of the graph the system is based on and its embedding. In these cases each spin is element of the support of exactly two star and plaquette operators (since every edge of an embedded multi graph has exactly two adjacent vertices and faces). Hence it holds $\prod_s A_s = \mathbb{1} = \prod_p B_p$. If one drops this restriction Toric Code systems with *open boundaries* can be constructed. In ref. [28] a system with z- and x-type boundaries is considered. In the following two systems with z-type boundaries are presented for explanatory purposes.

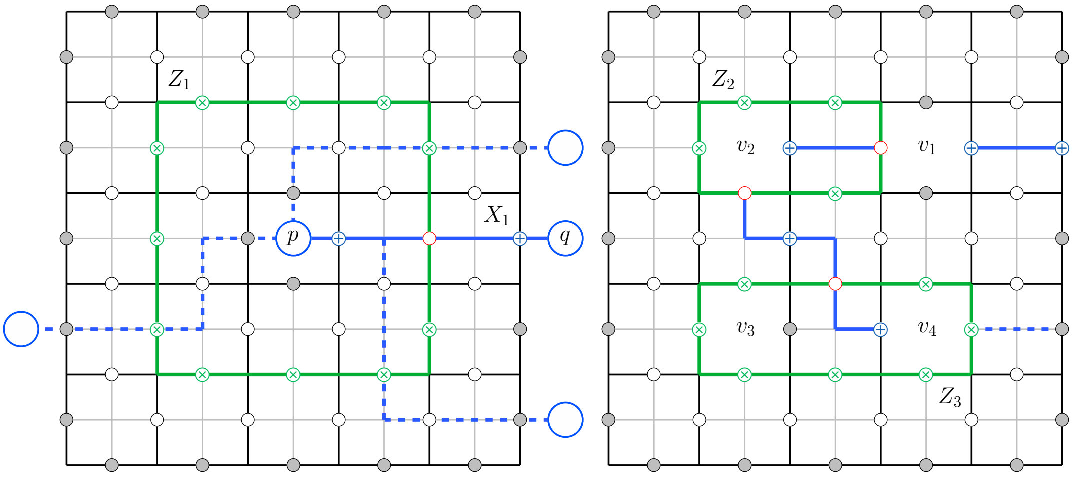


Figure 1.10 (Color online) : Left picture: A TCM with two disjoint z-type boundaries (grey spins). The protected space is 2-fold degenerate and the topological quantum number is measured by a z-type loop operator (green string) winding once around the "hole". Right picture: A system with 5 disjoint z-boundaries, thus encoding 4 logical qubits v_i, each identified with one of the "holes".

[14] If there is no open boundary as described in [28] the relation $\prod_s A_s = \mathbb{1} = \prod_p B_p$ holds.

1. Consider the Toric Code system depicted in fig. 1.10 on the left-hand side. This system features a “hole” where no plaquette operator B_p is available. Furthermore there are no plaquette operators surrounding the lattice. Hence the dual lattice $\mathcal{L}^*$ (light grey lattice) exhibits “open ends” at the grey marked spins. These spins form two disjoint z-boundaries (one around the hole and one around the whole system). However, there are no open ends attached to $\mathcal{L}$ (black lattice), thus there is no x-type boundary. Therefore we find $\prod_s A_s = \mathbb{1}$ but $\prod_p B_p \neq \mathbb{1}$. In order to compute the dimension of $\mathcal{PS}$ let N be the width (and height) of the system in numbers of plaquettes. Consequently there are $2N^2 + 2N$ spins and $N^2 + 2N + 1$ (dependent!) star operators. Furthermore there are N^2-1 (independent!) plaquette operators. Using Proposition 1.2.2 we find $\dim \mathcal{PS} = 2^{2N^2+2N-(N^2+2N+1+N^2-1-1)} = 2^1 = 2$. Thus one logical qubit may be encoded in this system.

   How this is done is illustrated in the picture. Imagine the x-type string operator X_1 is applied to the (unique) ground state $|\Psi\rangle$ of the same system *with* a plaquette operator filling the hole (which is of course also a ground state of the present system). Obviously $X_1 |\Psi\rangle$ is stabilized by $\mathcal{S}$ defined by the depicted system since there is no plaquette operator which could “detect” the m-type particles p and q attached to the endpoints of such a x-type string. Furthermore one verifies easily that the string defining X_1 may be deformed by applying star operators yielding strings denoted by dashed blue lines, for instance. Thus such a x-type string is “pinned” at z-type boundaries and may be “pulled” along the latter. While $|\Psi\rangle$ denotes the state that belongs to $v_{top} = +1$, $X_1 |\Psi\rangle$ is the second basis state with negative parity $v_{top} = -1$. Both states are distinguished measuring the z-type loop operator Z_1 winding once around the hole. One finds $Z_1 |\Psi\rangle = |\Psi\rangle$ and $Z_1 X_1 |\Psi\rangle = - |\Psi\rangle$.

   Note that the algebra generated by z-type loop operators and x-type string operators of the considered kind demands a more general notion of homology[15] on lattices than described in subsec. 1.4.1.

2. The system on the right-hand side of fig. 1.10 extents the notion of “piercing holes” in the system. Here 4 plaquette operators were deleted; thus the system features 5 disjoint z-boundaries and no x-boundary. Since there is no relation regarding plaquette operators that could be eliminated each new hole removes one stabilizer generator from $\mathcal{S}$. Hence the protected space of systems with m plaquette holes (and no plaquette surrounding the whole system) is 2^m-fold degenerate.

   It is reasonable to identify each of the m logical qubits encoded in such systems with one of the holes as illustrated in the figure. Then a topological quantum number v_i equals $+1$ (-1) if there is an even (odd) number of x-type strings attached to the corresponding z-boundary. Thus v_i is the eigenvalue of a z-type loop operator winding once around the corresponding hole. E.g. Z_2 returns the quantum number v_2 (here $v_2 = +1$) whereas Z_3 returns the *product* $v_3 v_4$ since it encloses two holes (here $v_3 v_4 = (+1)(-1) = -1$). Finally note that a combination of string operators acts trivially on $\mathcal{PS}$ if it forms a set of closed x-type loops *together* with the z-boundaries. Thus the combination of x-type string operators defined by the continuous and dashed blue lines equals $\mathbb{1}$, which shows that the continuous strings may be replaced by the single dashed one.

The examples given above feature z-boundaries only. Generally there may be z- and x-boundaries present in the same system increasing the degeneracy further. If there are at least two disjoint boundaries of each type the existence of non-trivial x- and z-type string operators acting on the protected space follows; thus the number of orthonormal ground states increases.

[15] Relative homology, see ref. [28] for further explanations.

1.6 Graph states

Graph states are a special class of stabilizer states introduced by Hein *et al.* [2] that can be described by simple graphs in terms of graph theory. Using graph states a variety of multi-qubit states in the field of quantum error correction, one-way quantum computing and quantum communication can be described not only in the framework of stabilizer codes but by means of graph theory. As it has been shown several physical properties regarding graph states may be derived in terms of structural properties of the corresponding graphs. Furthermore graph states turned out to be universal standard form of stabilizer states[16] since there is an LC-equivalent graph state for every stabilizer state (see sec. 1.7 below).

There are two equivalent definitions for graph states. The first one defines them as the unique state stabilized by a stabilizer group $\mathcal{S}[G]$ derived from a given simple graph G. The second approach states an algorithm to construct graph states from scratch using G as "construction plan". Both definitions will be outlined in the following. For a detailed introduction and further applications see ref. [8].

1.6.1 Definition

Definition 1.6.1: (Graph states, Construction). *Let G be an arbitrary simple graph with edges $E(G)$ and vertices $V(G)$. Consider a system of $N = |G|$ qubits (each identified with a vertex $v \in V(G)$) and the special phase gate*

$$U_{ab} := \mathcal{P}_a^{z,+} \otimes \mathbb{1}_b + \mathcal{P}_a^{z,-} \otimes \sigma_b^z \tag{1.63}$$

where $\mathcal{P}_v^{z,\pm} := \frac{1}{2}\left[\mathbb{1}_v \pm \sigma_v^z\right]$ are the common projectors onto $\{|0\rangle, |1\rangle\}$ operating on qubit v. Then the state vector

$$|G\rangle := \prod_{\{a,b\}\in E(G)} U_{ab} \left|+\right\rangle^{V(G)} \in \mathcal{H}^N \tag{1.64}$$

*is the **graph state** described by G. Here the notation $\left|+\right\rangle^{V(G)} := \bigotimes_{v\in V(G)} \left|+\right\rangle_v$ is used.*

It is easy to show that $U_{ab}^2 = \mathbb{1}$ and $U_{ab}^\dagger = U_{ab}$, thus applying U_{ab} adds an "edge" between vertices a and b modulo 2. Definition 1.6.1 is useful to construct graph states from scratch (see Example 1). However, to keep track of graph states transformed under Clifford operations a description in terms of stabilizers is advantageous as stated by the Gottesman-Knill theorem 1.2.3. Hence the following (alternative) definition of graph states, which can be shown [8] to be equivalent to def. 1.6.1:

Definition 1.6.2: (Graph states, Stabilizer). *Under the same conditions as stated in Lemma 1.6.1 define the following (independent) operators*

$$K_G^{(v)} := \sigma_v^x \prod_{w\in N_v} \sigma_w^z \in G_N \tag{1.65}$$

*Let $\mathcal{G} = \left\{K_G^{(v)}\right\}_{v\in V(G)}$. Then we call $\mathcal{S} = \operatorname{span}\mathcal{G}$ the **graph state stabilizer** derived from G. It is $\operatorname{rank}\mathcal{S} = |\mathcal{G}| = N$ and by Proposition 1.2.2 it follows $\dim \mathcal{PS} = 1$. The unique state $|G\rangle$, such that $\mathcal{PS} = \operatorname{span}\{|G\rangle\}$, is called the **graph state** described by G.*

The following example illustrates both definitions:

[16] Therefore some authors call stabilizer states (also known as *additive quantum codes*) *graph states*.

Example 1: (Graph states). Consider the 3-vertex star graph G depicted in fig. 1.11.

By def. 1.6.1 we obtain the corresponding graph state

$$\begin{aligned}|G\rangle &= U_{23}U_{12}\,|+\rangle_3\,|+\rangle_2\,|+\rangle_1 \equiv U_{23}U_{12}\,|+++\rangle \in \mathcal{H}^3 \\ &= \frac{1}{\sqrt{2}}U_{23}\,[|++0\rangle + |+-1\rangle] \\ &= \frac{1}{\sqrt{4}}\,[|{+00}\rangle + |{-10}\rangle + |{+01}\rangle - |{-11}\rangle] \\ &= \frac{1}{\sqrt{2}}\,[|{+0+}\rangle + |{-1-}\rangle] = H_1H_3\,|\mathrm{GHZ}_3\rangle\end{aligned}$$

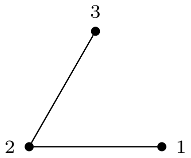


Figure 1.11 (Color online) : A simple star graph with 3 vertices.

Here H_v denotes a Hadamard gate operating on qubit v. It can be shown that any star graph represents a GHZ state up to some Hadamard transformations. The stabilizer generators derived from G are $\mathcal{G} = \{\sigma_1^x\sigma_2^z, \sigma_2^x\sigma_1^z\sigma_3^z, \sigma_3^x\sigma_2^z\}$. Clearly $g\,|G\rangle = |G\rangle$ for all $g \in \mathcal{G}$, thus $\mathcal{S}\,|G\rangle = |G\rangle$.

The intense study of graph states during the last years is justified by their simple description. Several properties and transformations may be expressed in purely graph theoretic terms. For instance, it was shown [2] that for special states the *Schmidt measure*, an entanglement measure [29] used for multipartite entanglement, can be computed from geometric properties of the underlying graphs. It was furthermore shown that local Pauli measurements[17] can be expressed as transformations of the corresponding graphs up to local unitaries.

One of the most interesting facts about graph states is the link between local Clifford operations and the corresponding transformations of their graphs. The so called *LC-rule* will be used throughout this thesis and is explained in detail in subsec. 1.6.2 below.

1.6.2 Local Clifford operations

Let $|G\rangle$ and $|G'\rangle$ denote two graph states of N qubits. We call them *equivalent under local Clifford operations* (*LC-equivalent*) iff there exists $C \in \mathcal{C}_N^l$ such that

$$|G\rangle = C\,|G'\rangle \tag{1.66}$$

and write $|G\rangle \sim_{\mathrm{LC}} |G'\rangle$. Note: Whereas the class of stabilizer states is closed under local Clifford transformations[18] this is not true for graph states in general. As stated in [2, 5, 8] the following proposition, known as *LC-rule*, can be used for the description of local Clifford equivalent graph states:

Proposition 1.6.3: (LC-rule). *Let $|G\rangle$ and $|G'\rangle$ be two graph states with corresponding graphs $G = (E, V)$ and $G' = (E', V)$. Then it holds*

$$|G\rangle \sim_{\mathrm{LC}} |G'\rangle \;\Leftrightarrow\; \exists\,\tau \in \mathcal{T} \,:\, G = \tau G' \;\Leftrightarrow\; G \sim_{\mathrm{LC}} G' \tag{1.67}$$

where $\mathcal{T} := \operatorname{span}\{\tau_a : \mathcal{E}(V) \to \mathcal{E}(V)\,|\, a \in V\}$ denotes the group of all local complementations τ acting on the edge space $\mathcal{E}(V)$. In short: Two graph states are LC-equivalent if and only if their graphs are connected by a sequence of local complementations.

For a proof see ref. [5]. The *LC-rule* will be used frequently in the following chapters and should be kept in mind.

The reader may notice that a fortunate coincidence leads to the same abbreviations for graph states equivalent under **Local Clifford operations** ($|G\rangle \sim_{\mathrm{LC}} |G'\rangle$) and their graphs equivalent under **Local Complementations** ($G \sim_{\mathrm{LC}} G'$). Hence the meaning of "LC-equivalent" depends on the context. However, since in most cases the graph G and its graph state $|G\rangle$ are identified these different meanings merge and "LC-equivalent" conveys both notions.

The following example illustrates the statement of Proposition 1.6.3:

[17] Projective measurements described by $\mathcal{P}_v^{x,\pm}$, $\mathcal{P}_v^{y,\pm}$ and $\mathcal{P}_v^{z,\pm}$.

[18] More generally stabilizer states are mapped onto stabilizer states under Clifford operations $\mathcal{C}_N$.

Example 2: (LC-rule). Consider the graph G depicted in fig. 1.12 on the left. The remaining three graphs represent LC-equivalent graph states since they are related by local complementations. Here the sequence $\tau_2\tau_6\tau_1$ was applied in order to obtain the graph on the right-hand side.

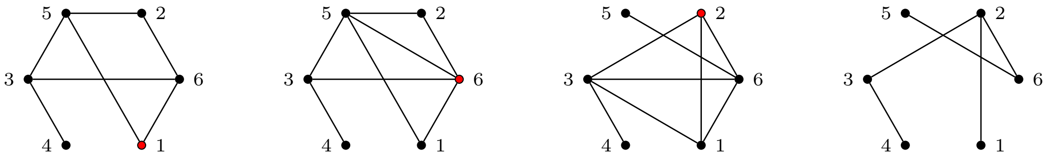


Figure 1.12 (Color online) : Successive application of the LC-rule. Starting on the left-hand side applying τ_v at the red marked vertex yields the next graph. This example will be used in subsec. 3.2.1.

1.7 Local unitary equivalence

The notion of equivalence under certain local operations is generalised by the following

Definition 1.7.1: (LU- and LC-equivalence). *Let $|\Psi\rangle$ and $|\Psi'\rangle$ be two arbitrary stabilizer states of N qubits. Then we call them*

*(i) equivalent under local unitary operations (**LU-equivalent**) iff there exists $U \in U(2)^{\otimes N}$ such that $|\Psi\rangle = U|\Psi'\rangle$. Write $|\Psi\rangle \sim_{\text{LU}} |\Psi'\rangle$.*

*(ii) equivalent under local Clifford operations (**LC-equivalent**) iff there exists $C \in \mathcal{C}_N^l$ such that $|\Psi\rangle = C|\Psi'\rangle$. Write $|\Psi\rangle \sim_{\text{LC}} |\Psi'\rangle$.*

Since LC-operations are local unitaries, $\mathcal{C}_N^l < U(2)^{\otimes N}$, it follows immediately that LC-equivalent states are LU-equivalent as well. For $\mathcal{C}_N^l \neq U(2)^{\otimes N}$ the converse cannot be true in general. However, due to the lack of counterexamples it was conjectured [30] that *for stabilizer states* LU- and LC-equivalence coincide. This statement, denoted by LU ⇔ LC, was called the LU-LC-conjecture. If it holds the impact on the description of stabilizer states and their properties would be formidable. The reason is given by the following proposition:

Proposition 1.7.2: (Stabilizer and graph states). *Let $|\Psi\rangle$ be an arbitrary stabilizer state of N qubits. Then there exists a graph state $|G\rangle$ and a local Clifford operation $C \in \mathcal{C}_N^l$ such that $|\Psi\rangle = C|G\rangle$. Hence graph states are universal standard forms of stabilizer states.*

A proof is given in [5] or more generally in [31]. Combining the LU-LC-conjecture and Proposition 1.7.2 one finds: First, if LU ⇔ LC (especially LU ⇒ LC) holds for graph states it holds for arbitrary stabilizer states as well. Secondly, to check whether two given stabilizer states are LU-equivalent becomes efficiently computable since it is sufficient to probe whether the graphs of their LC-equivalent graph states are related by a sequence of local complementations.

Although it was shown that LU ⇔ LC holds for a large class of stabilizer states [32, 33] the LU-LC-conjecture was disproved by Ji *et al.* [34] in 2009. Hence in general one cannot classify stabilizer states with respect to local unitaries by means of Proposition 1.6.3. Nevertheless it was shown in ref. [2] that LU ⇔ LC holds for graph states with up to 7 qubits. Furthermore it is believed [34] (but not proved) that the counterexample used in ref. [34] is the smallest ($N = 27$) one possible.

Recently the class of stabilizer states for which the LU-LC-conjecture holds was enlarged [27], including a large class of Toric Code Models. However, since most Toric Code systems considered later on do not belong to this class, only LC-classes will be considered if $N > 7$. Since all systems subject to this thesis have less than 27 spins it is strongly believed that all propositions stated for LC-operations hold for LU-operations as well.

Chapter 2

Ground states for generalised Toric Code Models

2.1 Objective and overview

The primary purpose of this chapter is to investigate how the topology of a graph $\mathcal{L}$ affects the structure of the Toric Code ground states stabilized by $\mathcal{S}[\mathcal{L}]$. Thus our main objective is the analysis of small spin settings ruled by the Toric Code Hamiltonian

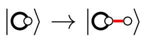
$$H_{TCM} = -\sum_{s\in V(\mathcal{L})} A_s - \sum_{p\in P(\mathcal{L})} B_p \tag{2.1}$$

where $\mathcal{L}$ is an arbitrary (multi) graph with faces $P(\mathcal{L})$ defined by a given embedding on a compact 2-manifold or by definition (see sec. 1.5). The ground states of this Hamiltonian form a basis of the subspace $\mathcal{PS}$ which is invariant under $\mathcal{S}[\mathcal{L}]$.

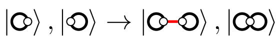
Since our ultimate goal is the analysis of small spin systems under the regime of a Toric Code stabilizer (see chapters 3 and 4), it might be useful to understand the transformation rules of special ground state vectors. With "special" we mean the basis vectors of the protected subspace that simultaneously diagonalize the closed z-type *loop operators* Z_i. Therefore the topological quantum numbers defined by their eigenvalues are good quantum numbers distinguishing the orthonormal ground states.

The transformations we want to analyze concern certain manipulations of the stabilizer topology. These elementary transformations are sufficient to construct arbitrary stabilizer structures including the classic Toric Code defined on a square lattice embedded into the torus. In particular the following elementary transformations are examined:

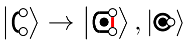
- Adding an edge at an arbitrary vertex of a given stabilizer:



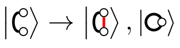
- Connecting two stabilizers via additional edges or a common vertex:



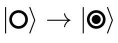
- Creating a new loop with and without a plaquette:



respectively



- Adding a plaquette



Obviously the ground state of a given stabilizer cannot determine the ground state after the transformation uniquely, if this transformation creates new homologously non-trivial paths.

2.2 Constructing TCM ground states

Adding an edge Consider the simplest case, extending a given stabilizer by attaching a new edge with a new A_s stabilizer generator at its end to an arbitrary vertex, that is.

To motivate the lemma stated below consider the (unique) ground state $|{\circ\!-\!\circ\!-\!\circ}\rangle$ without any plaquette operators. One finds

$$|{\circ\!-\!\circ\!-\!\circ}\rangle = \frac{1}{\sqrt{4}}\left[|00\rangle + |01\rangle + |10\rangle + |11\rangle\right]$$

Now calculate the ground state $|{\circ\!-\!\circ\!-\!\circ\!-\!\circ}\rangle$ of the stabilizer $\mathcal{S}'$ obtained by adding an edge to the end of the string. An easy calculation yields

$$\begin{aligned} |{\circ\!-\!\circ\!-\!\circ\!-\!\circ}\rangle &= \frac{1}{\sqrt{8}}\left[|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle\right] \\ &= \frac{1}{\sqrt{4}}\left[|00\rangle + |01\rangle + |10\rangle + |11\rangle\right] \otimes \frac{1}{\sqrt{2}}\left[|0\rangle + |1\rangle\right] \end{aligned}$$

and consequently

$$|{\circ\!-\!\circ\!-\!\circ\!-\!\circ}\rangle = |{\circ\!-\!\circ\!-\!\circ}\rangle \otimes |{\circ\!-\!\circ}\rangle$$

where $|{\circ\!-\!\circ}\rangle = |+\rangle$ as defined below. This observation holds true for arbitrary TCM ground states extended by a single edge (or spin) as the following lemma states:

Lemma 2.2.1. *Let $|{\circlearrowleft}\rangle$ be a ground state of an arbitrary TCM stabilizer $\mathcal{S}$ and let*

$$|{\circ\!-\!\circ}\rangle := |+\rangle = \frac{1}{\sqrt{2}}\left[|0\rangle + |1\rangle\right] \tag{2.2}$$

be the TCM ground state of a single spin on a 2-vertex graph. Then the ground state $|{\circlearrowleft\!-\!\circ}\rangle$ of a new stabilizer $\mathcal{S}'$ where a single edge is added at an arbitrary vertex q is given by

$$|{\circlearrowleft\!-\!\circ}\rangle = |{\circlearrowleft}\rangle \otimes |{\circ\!-\!\circ}\rangle \tag{2.3}$$

and the degeneracy of the new protected subspace is $\gamma_{\mathcal{S}'} = \gamma_{\mathcal{S}}$ where $\gamma_{\mathcal{S}} \equiv \dim \mathcal{PS}$.

Proof. It is straightforward to show that $\gamma_{\mathcal{S}'} = \gamma_{\mathcal{S}}$ since this transformation adds both one new spin and one new stabilizer generator (the A_s operator at the end of the string) therefore leaving the degeneracy unchanged. Now let

$$|{\circlearrowleft\!-\!\circ}\rangle = |{\circlearrowleft}\rangle \otimes |{\circ\!-\!\circ}\rangle$$

Since no new B_p is added all generators B'_p act solely on the state $|{\circlearrowleft}\rangle$. Consequently $B'_p\,|{\circlearrowleft\!-\!\circ}\rangle = |{\circlearrowleft\!-\!\circ}\rangle\ \forall_{B'_p \in \mathcal{S}'}$. Obviously $A'_s\,|{\circlearrowleft\!-\!\circ}\rangle = |{\circlearrowleft\!-\!\circ}\rangle$ for all $A'_s = A_s \otimes \mathbb{1} \in \mathcal{S}'$ acting trivially or $A'_s = \mathbb{1} \otimes \sigma^x \in \mathcal{S}'$ solely on the new spin since $\sigma^x\,|{\circ\!-\!\circ}\rangle = |{\circ\!-\!\circ}\rangle$. The only critical star operator A'_q sits at vertex q where the new edge was attached. This operator can be written as tensor product $A'_q = A_q \otimes \sigma^x$ and consequently $A'_q\,|{\circlearrowleft\!-\!\circ}\rangle = |{\circlearrowleft\!-\!\circ}\rangle$. z-type loop operators are diagonal over $|{\circlearrowleft}\rangle$ by assumption and therefore over $|{\circlearrowleft\!-\!\circ}\rangle$ since there is no *closed* string operator on the edges of $\mathcal{L}'$ acting non-trivially on the attached spin. ■

Connecting two stabilizers An interesting question might be how to connect two given stabilizers (more precisely: their corresponding graphs) by a common vertex or a string of arbitrary length and how the new ground states can be derived from the old ones.

To get a notion of this procedure let $|{\blacktriangleright\!\!\circ}\rangle$ be the (unique) ground state of a triangle with a plaquette filling its face and let $|{\triangleright\!\!\circ}\rangle_{-}$ denote one of the two ground states of the same triangle without this plaquette (therefore we may choose -1 for the product of spins surrounding the triangle). One computes

$$\begin{aligned} |\blacktriangleright\rangle &= \frac{1}{\sqrt{4}}\left[|000\rangle + |011\rangle + |101\rangle + |110\rangle\right] \\ |\triangleright\rangle_- &= \frac{1}{\sqrt{4}}\left[|001\rangle + |010\rangle + |100\rangle + |111\rangle\right] \end{aligned}$$

Let these two triangles share a common edge and compute the ground state $|\blacktriangleright\!\triangleleft\rangle_-$ with spin product -1 around the open loop. This computation yields

$$\begin{aligned} |\blacktriangleright\!\triangleleft\rangle_- = \frac{1}{\sqrt{16}} \quad & [|001000\rangle + |001011\rangle + |001101\rangle + |001110\rangle + |010000\rangle + |010011\rangle + \\ & |010101\rangle + |010110\rangle + |100000\rangle + |100011\rangle + |100101\rangle + |100110\rangle + \\ & |111000\rangle + |111011\rangle + |111101\rangle + |111110\rangle] \end{aligned}$$

Obviously it holds

$$|\blacktriangleright\!\triangleleft\rangle_- = |\triangleright\rangle_- \otimes |\blacktriangleright\rangle$$

and if $|\triangleright\rangle_+$ is considered instead of $|\triangleright\rangle_-$ it follows $|\blacktriangleright\!\triangleleft\rangle_+ = |\triangleright\rangle_+ \otimes |\blacktriangleright\rangle$ by flipping one of the spins around the open loop.

This simple example leads to the conjecture that the ground states of the one-vertex connection of two Toric Code systems are given by the tensor product of the original ground states.

Lemma 2.2.2. *Let $|\circlearrowright\rangle$ and $|\circlearrowleft\rangle$ be two TCM ground states stabilized by $\mathcal{S}_1$ and $\mathcal{S}_2$. Then the one-vertex connection at vertices q_1 and q_2 has an uniquely defined ground state $|\infty\rangle$ given by*

$$|\infty\rangle = |\circlearrowright\rangle \otimes |\circlearrowleft\rangle \tag{2.4}$$

Alternatively one may connect the two systems by means of a N-edge string. In this case for the new ground state it follows

$$|\circlearrowright\!-\!\circlearrowleft\rangle = |\circlearrowright\rangle \otimes |\circ\!-\!\circ\rangle^{\otimes N} \otimes |\circlearrowleft\rangle \tag{2.5}$$

where $|\circ\!-\!\circ\rangle^{\otimes N} = \bigotimes_{i=1}^{N} |+\rangle_i$ denotes the ground state of a N-spin string. The new degeneracy is $\gamma_{\mathcal{S}'} = \gamma_{\mathcal{S}_1} \cdot \gamma_{\mathcal{S}_1}$.

Proof. That $\gamma_{\mathcal{S}'} = \gamma_{\mathcal{S}_1} \cdot \gamma_{\mathcal{S}_1}$ can be seen easily by straightforward calculation

$$\begin{aligned} \gamma_{\mathcal{S}'} &= 2^{e'-(v'+f'-c')} \\ &= 2^{(e_1+e_2+N)-((v_1+v_2+N-1)+(f_1+f_2)-(c_1+c_2-1))} \\ &= 2^{e_1-(v_1+f_1-c_1)} \cdot 2^{e_2-(v_2+f_2-c_2)} \\ &= \gamma_{\mathcal{S}_1} \cdot \gamma_{\mathcal{S}_1} \end{aligned}$$

where e, v and f denote the numbers of edges, vertices and faces. c is the number of constraints for star and plaquette operators. Since the newly added edge does not interfere with possible constraints for B_p operators it may only change constraints regarding A_s operators. Since both stabilizers $\mathcal{S}_{1,2}$ featured such a constraint there is a loss of one constraint due to the A_s-link between the two stabilizers.

Let now

$$|\circlearrowright\!-\!\circlearrowleft\rangle = |\circlearrowright\rangle \otimes |\circ\!-\!\circ\rangle^{\otimes N} \otimes |\circlearrowleft\rangle \quad \text{and} \quad |\infty\rangle = |\circlearrowright\rangle \otimes |\circlearrowleft\rangle$$

Since no new B_p is added all generators B'_p act solely on one of the two states $|\circlearrowright\rangle$ or $|\circlearrowleft\rangle$ and consequently $B'_p\,|\circlearrowright\!-\!\circlearrowleft\rangle = |\circlearrowright\!-\!\circlearrowleft\rangle\ \forall_{B'_p \in \mathcal{S}'}$. An analogue argument shows that $A'_s\,|\circlearrowright\!-\!\circlearrowleft\rangle = |\circlearrowright\!-\!\circlearrowleft\rangle\ \forall_{A'_s = A_{s_1} \otimes \mathbb{1},\, A'_s = \mathbb{1} \otimes A_{s_2} \in \mathcal{S}'}$. That the A_s operators acting on changed or added vertices (A_{q_1} and A_{q_2}) act trivially on $|\circlearrowright\!-\!\circlearrowleft\rangle$ can be understood by the same argument as presented in the proof of Lemma 2.2.1. Let $Z\left[\mathcal{C}\right]$ be a z-type loop operator with *closed* path $\mathcal{C}$ on $\mathcal{L}'$. If $Z\left[\mathcal{C}\right]$ acts trivially on one of the two components $|\bigcirc\!\!\circ\rangle$ or $|\circ\!\!\bigcirc\rangle$ it follows immediately $Z\left[\mathcal{C}\right]|\bigcirc\!\!\circ\!\!-\!\!\circ\!\!\bigcirc\rangle = \pm\,|\bigcirc\!\!\circ\!\!-\!\!\circ\!\!\bigcirc\rangle$ due to assumption. If $\mathcal{C}$ crosses the new established link between the two components this occurs an even number of times since this link is the *only* one. Therefore $\mathcal{C}$ decomposes into two closed paths $\mathcal{C}_1$ and $\mathcal{C}_2$ each on one of the two spin systems. Hence

$$Z\left[\mathcal{C}\right]|\bigcirc\!\!\circ\!\!-\!\!\circ\!\!\bigcirc\rangle = Z\left[\mathcal{C}_1\right]|\bigcirc\!\!\circ\rangle \otimes |\circ\!\!-\!\!\circ\rangle^{\otimes N} \otimes Z\left[\mathcal{C}_2\right]|\circ\!\!\bigcirc\rangle = \pm\,|\bigcirc\!\!\circ\!\!-\!\!\circ\!\!\bigcirc\rangle$$

■

Remark 2.2.3. *Obviously Lemma 2.2.1 can be derived from Lemma 2.2.2 as a special case.*

Creating a loop After these rather technical investigations consider the more interesting case where a new loop is created by reconnecting a stabilizer by means of a string to itself thus creating a new loop in the corresponding graph. As before we start with a simple example to establish the notion of the general statement following.

Let $|\circ\!\!-\!\!\circ\!\!-\!\!\circ\rangle$ be the unique ground state of two spins on a string (as used above)

$$|\circ\!\!-\!\!\circ\!\!-\!\!\circ\rangle = \frac{1}{\sqrt{4}}\left[|00\rangle + |01\rangle + |10\rangle + |11\rangle\right]$$

Add a spin to the system as an edge of a new stabilizer $\mathcal{S}'$ connecting the first with the last vertex of the string, hence creating a loop (here: a triangle). We already know the two possible ground states of the new stabilizer

$$\begin{aligned} |\triangleright\!\!\circ\rangle_+ &= \frac{1}{\sqrt{4}}\left[|000\rangle + |011\rangle + |101\rangle + |110\rangle\right] \\ |\triangleright\!\!\circ\rangle_- &= \frac{1}{\sqrt{4}}\left[|001\rangle + |010\rangle + |100\rangle + |111\rangle\right] \end{aligned}$$

These ground states may be rewritten

$$\begin{aligned} |\triangleright\!\!\circ\rangle_+ &= \frac{1}{\sqrt{4}}\left[(|00\rangle + |11\rangle)\otimes|0\rangle + (|01\rangle + |10\rangle)\otimes|1\rangle\right] = \frac{1}{\sqrt{2}}\left[\left|\Phi^+\right\rangle_o \otimes |0\rangle_n + \left|\Psi^+\right\rangle_o \otimes |1\rangle_n\right] \\ |\triangleright\!\!\circ\rangle_- &= \frac{1}{\sqrt{4}}\left[(|01\rangle + |10\rangle)\otimes|0\rangle + (|00\rangle + |11\rangle)\otimes|1\rangle\right] = \frac{1}{\sqrt{2}}\left[\left|\Psi^+\right\rangle_o \otimes |0\rangle_n + \left|\Phi^+\right\rangle_o \otimes |1\rangle_n\right] \end{aligned}$$

and we find that entanglement between the new spin (n) and the old system (o) occurs. Here $|\Psi^\pm\rangle$ and $|\Phi^\pm\rangle$ denote the Bell states. This observation gives rise to the following

Assumption 2.2.4. *Multipartite entanglement in TCM ground states is an inherent consequence of cycles in the corresponding graph.*

Indeed, by Lemma 2.2.1 it follows that every stabilizer without any cycle is an unentangled product state of spins since its corresponding graph is a tree in terms of graph theory and therefore may be constructed exclusively by concatenating edges without forming loops. In case of our current example this reads

$$|\circ\!\!-\!\!\circ\!\!-\!\!\circ\rangle = |\circ\!\!-\!\!\circ\rangle \otimes |\circ\!\!-\!\!\circ\rangle = |+\rangle^{\otimes 2}$$

In contrast $|\triangleright\!\!\circ\rangle_\pm$ shows real multipartite entanglement as one computes easily in the Bell basis $\{|\Phi^\pm\rangle, |\Psi^\pm\rangle\}$

$$\rho_i = \mathrm{Tr}_{E(\mathcal{L})\setminus i}\left[|\triangleright\!\!\circ\rangle\langle\triangleright\!\!\circ|_\pm\right] = \frac{1}{2}\left[|0\rangle\langle 0|_i + |1\rangle\langle 1|_i\right] = \frac{1}{2}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}_i$$

for an arbitrary spin i. Hence $\mathrm{Tr}\left[\rho_i^2\right] = \frac{1}{2} < 1$. Here $\mathrm{Tr}_A\left[\cdot\right]$ denotes the partial trace with respect to A.

Let us return to the new ground states and how they can be constructed from previous ones. Note that they can be rewritten as follows

$$\begin{aligned}\left|\text{▷}\right\rangle_{\pm} &= \frac{1}{2}\left(\mathbb{1} \pm \sigma_1^z \otimes \sigma_2^z\right)\left|\text{○–○–○}\right\rangle_{1,2} \otimes |0\rangle_3 + \frac{1}{2}\left(\mathbb{1} \mp \sigma_1^z \otimes \sigma_2^z\right)\left|\text{○–○–○}\right\rangle_{1,2} \otimes |1\rangle_3 \\ &= \mathcal{P}^{\pm}\left[\mathcal{C}_{pq}\right]\left|\text{○–○–○}\right\rangle_{1,2} \otimes |0\rangle_3 + \mathcal{P}^{\mp}\left[\mathcal{C}_{pq}\right]\left|\text{○–○–○}\right\rangle_{1,2} \otimes |1\rangle_3\end{aligned}$$

where $\mathcal{P}^{\pm}\left[\mathcal{C}_{pq}\right] = \frac{1}{2}\left[\mathbb{1} \pm \bigotimes_{i\in\mathcal{C}_{pq}} \sigma_i^z\right]$ denotes the projector onto the $\pm$-eigenstates of a z-type string operator starting at the left vertex p of our 2-spin string and ending at the right one q. In fact, this rule holds true for arbitrary TCM stabilizers as stated by the following

Lemma 2.2.5. *Let $\left|\text{ᑕ}\right\rangle$ be a TCM stabilizer ground state with vertices p and q (denoted by circles). Let furthermore*

$$\mathcal{P}^{\pm}\left[\mathcal{C}_{pq}\right] := \frac{1}{2}\left(\mathbb{1} \pm Z\left[\mathcal{C}_{pq}\right]\right) \tag{2.6}$$

be the projectors onto the eigenstates of string operators

$$Z\left[\mathcal{C}_{pq}\right] = \bigotimes_{i\in\mathcal{C}_{pq}} \sigma_i^z \tag{2.7}$$

with path $\mathcal{C}_{pq}$ on $\mathcal{L}$ with endpoints p and q. Consider the stabilizer $\mathcal{S}'$ that corresponds to the graph obtained by connecting p and q by means of a single edge (thus adding one new spin). Then there are two orthogonal ground states $\left|\text{ᑕ|}\right\rangle_{\pm}$ of the new stabilizer $\mathcal{S}'$ given by[19]

$$\left|\text{ᑕ|}\right\rangle_{\pm} = \mathcal{P}^{\pm}\left[\mathcal{C}_{pq}\right]\left|\text{ᑕ}\right\rangle \otimes |0\rangle + \mathcal{P}^{\mp}\left[\mathcal{C}_{pq}\right]\left|\text{ᑕ}\right\rangle \otimes |1\rangle \tag{2.8}$$

and the degeneracy doubles $\gamma_{\mathcal{S}'} = 2\gamma_{\mathcal{S}}$.

Proof. That $\gamma_{\mathcal{S}'} = 2\gamma_{\mathcal{S}}$ follows since one spin but no additional stabilizer generator is added, therefore $\gamma_{\mathcal{S}'} = 2^{e+1-(v+f-c)} = 2\gamma_{\mathcal{S}}$. Now consider the new stabilizer $\mathcal{S}'$. Obviously $\left[\mathcal{P}^{\pm}\left[\mathcal{C}_{pq}\right], B'_p\right] = 0$ for all $p \in P(\mathcal{L}')$. Since the new spin is not affected by any plaquette operator B'_p it follows immediately $B'_p\left|\text{ᑕ|}\right\rangle_{\pm} = \left|\text{ᑕ|}\right\rangle_{\pm}$. The same argument holds for all A'_s where $s \notin \{p, q\}$ since $\left[A'_s, Z\left[\mathcal{C}_{pq}\right]\right] = 0$. If $s \in \{p, q\}$ it holds $\left\{A'_s, Z\left[\mathcal{C}_{pq}\right]\right\} = 0$ and it follows

$$\begin{aligned}A'_s\left|\text{ᑕ|}\right\rangle_{\pm} &= A'_s\mathcal{P}^{\pm}\left[\mathcal{C}_{pq}\right]\left|\text{ᑕ}\right\rangle \otimes |0\rangle + A'_s\mathcal{P}^{\mp}\left[\mathcal{C}_{pq}\right]\left|\text{ᑕ}\right\rangle \otimes |1\rangle \\ &= \mathcal{P}^{\mp}\left[\mathcal{C}_{pq}\right]A'_s\left|\text{ᑕ}\right\rangle \otimes |0\rangle + \mathcal{P}^{\pm}\left[\mathcal{C}_{pq}\right]A'_s\left|\text{ᑕ}\right\rangle \otimes |1\rangle \\ &= \mathcal{P}^{\mp}\left[\mathcal{C}_{pq}\right]\left|\text{ᑕ}\right\rangle \otimes |1\rangle + \mathcal{P}^{\pm}\left[\mathcal{C}_{pq}\right]\left|\text{ᑕ}\right\rangle \otimes |0\rangle \\ &= \left|\text{ᑕ|}\right\rangle_{\pm}\end{aligned}$$

Here we used $A'_s\mathcal{P}^{\pm}\left[\mathcal{C}_{pq}\right] = \mathcal{P}^{\mp}\left[\mathcal{C}_{pq}\right]A'_s$ for $s \in \{p, q\}$.

Consider the *closed* path $\mathcal{C}$ and its corresponding z-type string operator $Z[\mathcal{C}]$. Obviously $Z[\mathcal{C}]\left|\text{ᑕ|}\right\rangle_{\pm} = c_{\mathcal{C}}\left|\text{ᑕ|}\right\rangle_{\pm}$ if $Z[\mathcal{C}]$ acts trivially on the new spin and $c_{\mathcal{C}} \in \{1, -1\}$ denotes the eigenvalue with respect to $\left|\text{ᑕ}\right\rangle$. Let $\mathcal{C}^*$ be a *closed* path on $\mathcal{L}'$ with support on the newly added edge. The paths under consideration can be decomposed as follows

$$\mathcal{C}^* + \mathcal{C}_{pq} = \mathcal{C}^*_{pq} + \mathcal{C}' \quad \Rightarrow \quad \mathcal{C}^* = \mathcal{C}^*_{pq} + \mathcal{C}' + \mathcal{C}_{pq}$$

Here addition is taken modulo 2 and up to homology as explained in subsec. 1.4.1 (see fig. 2.1). $\mathcal{C}^*_{pq}$ denotes a *closed* path running through vertices p and q but acting trivially on the new spin (green path in fig. 2.1) and $\mathcal{C}'$ denotes the 1-edge path from p to q on the new spin.

[19] Up to a normalizing constant.

Hence it follows

$$\begin{aligned}
Z\left[\mathcal{C}^*\right]\left|\mathfrak{I}\right\rangle_\pm &= Z\left[\mathcal{C}^*_{pq}\right] Z\left[\mathcal{C}'\right] Z\left[\mathcal{C}_{pq}\right]\left|\mathfrak{I}\right\rangle_\pm \\
&= Z\left[\mathcal{C}^*_{pq}\right] Z\left[\mathcal{C}_{pq}\right]\mathcal{P}^\pm\left[\mathcal{C}_{pq}\right]\left|\mathfrak{C}\right\rangle\otimes\sigma^z\left|0\right\rangle + Z\left[\mathcal{C}^*_{pq}\right] Z\left[\mathcal{C}_{pq}\right]\mathcal{P}^\mp\left[\mathcal{C}_{pq}\right]\left|\mathfrak{C}\right\rangle\otimes\sigma^z\left|1\right\rangle \\
&= \pm\mathcal{P}^\pm\left[\mathcal{C}_{pq}\right] Z\left[\mathcal{C}^*_{pq}\right]\left|\mathfrak{C}\right\rangle\otimes\left|0\right\rangle \pm \mathcal{P}^\mp\left[\mathcal{C}_{pq}\right] Z\left[\mathcal{C}^*_{pq}\right]\left|\mathfrak{C}\right\rangle\otimes\left|1\right\rangle \\
&= \pm c_{\mathcal{C}^*_{pq}}\left\{\mathcal{P}^\pm\left[\mathcal{C}_{pq}\right]\left|\mathfrak{C}\right\rangle\otimes\left|0\right\rangle + \mathcal{P}^\mp\left[\mathcal{C}_{pq}\right]\left|\mathfrak{C}\right\rangle\otimes\left|1\right\rangle\right\} \\
&= \pm c_{\mathcal{C}^*_{pq}}\left|\mathfrak{I}\right\rangle_\pm
\end{aligned}$$

Where $c_{\mathcal{C}^*_{pq}} \in \{1,-1\}$ denotes the eigenvalue of $Z\left[\mathcal{C}^*_{pq}\right]$ with respect to $\left|\mathfrak{C}\right\rangle$. This completes the proof. ■

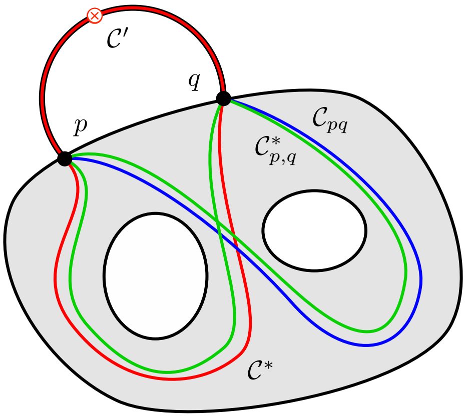


Figure 2.1 (Color online) : Adding paths as used in the proof of Lemma 2.2.5.

The calculation rules given by eq. (2.8) (or (2.11a), see below) are stated in a form that one grasps the transformation by intuition. For actual computations they may be simplified:

Remark 2.2.6. *Rewriting equation (2.8) or (2.11a) yields*

$$\left|\mathfrak{I}\right\rangle_\pm = \frac{1}{\sqrt{2}}\left[\left|\mathfrak{C}\right\rangle\otimes\left|+\right\rangle \pm Z\left[\mathcal{C}_{pq}\right]\left|\mathfrak{C}\right\rangle\otimes\left|-\right\rangle\right] \tag{2.9}$$

In fact, Lemma 2.2.5 is generalised to and subsequently derived from the following statement in combination with Lemma 2.2.1:

Lemma 2.2.7. *Let $\left|\mathfrak{C}\right\rangle$, p, q and $\mathcal{C}_{pq}$ be defined as in Lemma 2.2.5. Consider the stabilizer $\mathcal{S}'$ that corresponds to the graph obtained by connecting p and q directly. Then there are two orthogonal ground states $\left|\mathfrak{O}\right\rangle_\pm$ of the new stabilizer $\mathcal{S}'$ given by*[20]

$$\left|\mathfrak{O}\right\rangle_\pm = \sqrt{2}\cdot\mathcal{P}^\pm\left[\mathcal{C}_{pq}\right]\left|\mathfrak{C}\right\rangle \tag{2.10}$$

and the degeneracy doubles $\gamma_{\mathcal{S}'} = 2\gamma_{\mathcal{S}}$.

[20] Up to a normalizing constant.

Proof. That $\gamma_{\mathcal{S}'} = 2\gamma_{\mathcal{S}}$ follows from the removal of one independent star operator by merging the two operators A_p and A_q. One obtains $A'_s = A_p \otimes A_q$ for the new star operator. It follows

$A'_s\,|$ $\rangle_\pm = \sqrt{2}\cdot A'_s\mathcal{P}^\pm\left[\mathcal{C}_{pq}\right]|$ $\rangle$

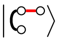
$= \sqrt{2}\cdot A_pA_q\mathcal{P}^\pm\left[\mathcal{C}_{pq}\right]|$ $\rangle$

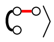
$= \sqrt{2}\cdot A_p\mathcal{P}^\mp\left[\mathcal{C}_{pq}\right]A_q\,|$ $\rangle$

$= \sqrt{2}\cdot \mathcal{P}^\pm\left[\mathcal{C}_{pq}\right]A_pA_q\,|$ $\rangle$

$= \sqrt{2}\cdot \mathcal{P}^\pm\left[\mathcal{C}_{pq}\right]|$ $\rangle = |$ $\rangle_\pm$

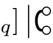
Let $\mathcal{C}$ be a path homologous to $\mathcal{C}_{pq}$ which is a *closed* path in the new system surrounding the created hole. Then it follows *by definition* of $\mathcal{P}^\pm\left[\mathcal{C}_{pq}\right]$ that $Z\left[\mathcal{C}\right]|$ $\rangle_\pm = \pm\,|$ $\rangle_\pm$. ■

As mentioned above one can now derive eq. (2.8) using Lemma 2.2.1, $|$ $\rangle \otimes |$ $\rangle = |$$\rangle$, that is. One obtains

$|$ $\rangle_\pm \overset{2.2.7}{=} \sqrt{2}\cdot\mathcal{P}^\pm\left[\mathcal{C}_{pq}+\mathcal{C}'\right]|$$\rangle$

$\overset{2.2.1}{=} \sqrt{2}\cdot\mathcal{P}^\pm\left[\mathcal{C}_{pq}+\mathcal{C}'\right]|$ $\rangle \otimes |$ $\rangle$

$= \dfrac{1}{\sqrt{2}}\left[\mathbb{1} \pm Z\left[\mathcal{C}_{pq}\right]Z\left[\mathcal{C}'\right]\right]|$ $\rangle \otimes |$ $\rangle$

$= \dfrac{1}{\sqrt{2}}\left[|\right.$ $\rangle \otimes |+\rangle \pm Z\left[\mathcal{C}_{pq}\right]|$$\left.\rangle \otimes |-\rangle\right]$

$\overset{2.2.6}{=} \mathcal{P}^\pm\left[\mathcal{C}_{pq}\right]|$ $\rangle \otimes |0\rangle + \mathcal{P}^\mp\left[\mathcal{C}_{pq}\right]|$ $\rangle \otimes |1\rangle$

Here $Z\left[\mathcal{C}'\right]|$ $\rangle = \sigma^z\,|+\rangle = |-\rangle$ was used.

Adding a plaquette Up to now we changed the stabilizer by adding various A_s operators without changing the subgroup generated by $\{B_p\}_{p\in P(\mathcal{L})}$. We saw in Lemma 2.2.5 that loops are the source of degeneracy and multipartite entanglement. What happens, if a new loop is created and a plaquette bounded by the new edge and $\mathcal{C}_{pq}$ is added simultaneously? Consider our example from above: The triangle. We found

$|$ $\rangle_+ = \dfrac{1}{\sqrt{4}}\left[|000\rangle + |011\rangle + |101\rangle + |110\rangle\right]$

$|$ $\rangle_- = \dfrac{1}{\sqrt{4}}\left[|001\rangle + |010\rangle + |100\rangle + |111\rangle\right]$

for the two possible ground states without constraints imposed by a plaquette. On the other hand the only allowed ground state of a filled triangle is

$|$ $\rangle = \dfrac{1}{\sqrt{4}}\left[|000\rangle + |011\rangle + |101\rangle + |110\rangle\right]$

Note that an additional plaquette does not *change* the ground states that where allowed without it but "chooses" the ones with topological quantum number $+1$ around the loop (the eigenvalue of $Z\left[\mathcal{C}\right]$ where $\mathcal{C}$ winds once around the hole).

Hence

$|$ $\rangle = |$ $\rangle_+$

This leads to the following

Lemma 2.2.8. *Let all conditions be the same as in Lemma 2.2.5 or 2.2.7 with the only difference that a new plaquette is added simultaneously (bounded by the new edge and $\mathcal{C}_{pq}$ or $\mathcal{C}_{pq}$ alone, respectively). Then there is only one new ground state $\left|\ \right\rangle$,$|\ \rangle$, given by*

$$\begin{aligned}\left|\ \right\rangle &= \mathcal{P}^{+}\left[\mathcal{C}_{pq}\right]\left|\ \right\rangle \otimes |0\rangle + \mathcal{P}^{-}\left[\mathcal{C}_{pq}\right]\left|\ \right\rangle \otimes |1\rangle && (2.11a)\\ |\ \rangle &= \sqrt{2}\cdot\mathcal{P}^{+}\left[\mathcal{C}_{pq}\right]\left|\ \right\rangle && (2.11b)\end{aligned}$$

and the degeneracy is left unchanged $\gamma_{\mathcal{S}'} = \gamma_{\mathcal{S}}$.

Proof. That $\left|\ \right\rangle$ is the new ground state satisfying all conditions follows immediately from Lemma 2.2.5. Especially

$$B'_p\left|\ \right\rangle = Z\left[\mathcal{C}_{pq}+\mathcal{C}'\right]\left|\ \right\rangle_{+} = Z\left[\mathcal{C}_{pq}\right]Z\left[\mathcal{C}'\right]\left|\ \right\rangle_{+} = +c_{\mathcal{C}^*_{pq}}\left|\ \right\rangle_{+} = \left|\ \right\rangle$$

Where B'_p denotes the added plaquette operator and $c_{\mathcal{C}^*_{pq}} = c_0 = +1$ since $\mathcal{C}^*_{pq} = 0$ in the first homology group[21]. That $\gamma_{\mathcal{S}'} = \gamma_{\mathcal{S}}$ follows since one spin and one stabilizer generator (B'_p) are added and there is no change in conditions reducing the number of generators. The proof for $|\ \rangle$ reads similarly and is left as an exercise to the reader. ■

Remark 2.2.9. *It follows immediately from 2.2.1, 2.2.5 and 2.2.8 that*

$$\left|\ \right\rangle \otimes |\ \rangle = \left|\ \right\rangle = \left|\ \right\rangle = \frac{1}{\sqrt{2}}\left[\left|\ \right\rangle_{+} + \left|\ \right\rangle_{-}\right] \qquad \text{and} \qquad \left|\ \right\rangle = \left|\ \right\rangle_{+} \tag{2.12}$$

Finally we need a transformation rule for a given ground state basis $\{|\mathrm{GS}_i\rangle\}_{i=1}^{\gamma_{\mathcal{S}}}$ if a plaquette B_{new} with arbitrary boundary $\mathcal{C}$ is added as a generator to $\mathcal{S}$. Obviously $B_{new} = Z[\mathcal{C}]$ by definition. This gives rise to the following statement:

Lemma 2.2.10. *Let $\mathcal{GS} = \{|\mathrm{GS}_i\rangle\}_{i=1}^{\gamma_{\mathcal{S}}}$ be the ONB of the protected subspace $\mathcal{PS} \leq \mathcal{H}^N$ defined by the stabilizer $\mathcal{S}$ with degeneracy $\gamma_{\mathcal{S}}$. Define a new stabilizer $\mathcal{S}' = \operatorname{span}\{\mathcal{S}\cup\{B_{new}\}\}$ where $B_{new} = Z[\mathcal{C}] \notin \mathcal{S}$ is a new plaquette operator. Then*

$$\mathcal{GS}' = \mathcal{P}^{+}[\mathcal{C}]\,\mathcal{GS}\setminus\{0\} \tag{2.13}$$

where $\mathcal{P}^{+}[\mathcal{C}]\,\mathcal{GS} := \{\mathcal{P}^{+}[\mathcal{C}]\,|\mathrm{GS}\rangle \mid |\mathrm{GS}\rangle \in \mathcal{GS}\}$ and the new degeneracy is $\gamma_{\mathcal{S}'} = \frac{\gamma_{\mathcal{S}}}{2}$.

Proof. Note that by assumption $\mathcal{P}^{\pm}[\mathcal{C}^*]$ is diagonal over $\mathcal{GS}$ for all closed paths $\mathcal{C}^*$ on $\mathcal{L}$ (with possible eigenvalues 0 and 1). Obviously $\mathcal{GS}'$ is an ONS. Furthermore $\mathcal{S}\,|\mathrm{GS}'\rangle = |\mathrm{GS}'\rangle$ for all $|\mathrm{GS}'\rangle \in \mathcal{GS}'$. That $B_{new}\,|\mathrm{GS}'\rangle = |\mathrm{GS}'\rangle$ follows by construction and $Z[\mathcal{C}^*]\,|\mathrm{GS}'\rangle = \pm|\mathrm{GS}'\rangle$ for all closed paths $\mathcal{C}^*$ by definition of $\mathcal{GS}$. Hence $\operatorname{span}\mathcal{GS}' \leq \mathcal{PS}'$. If $|\mathrm{GS}'\rangle \in \mathcal{PS}'$ it follows $|\mathrm{GS}'\rangle = \sum_{i=1}^{\gamma_{\mathcal{S}}}\alpha_i\,|\mathrm{GS}_i\rangle$ for some $\alpha_i \in \mathbb{C}$ since $\mathcal{PS}' \leq \mathcal{PS}$. It holds $|\mathrm{GS}'\rangle = \mathcal{P}^{+}[\mathcal{C}]\,|\mathrm{GS}'\rangle = \sum_{i=1}^{\gamma_{\mathcal{S}}}\alpha_i\mathcal{P}^{+}[\mathcal{C}]\,|\mathrm{GS}_i\rangle = \sum_{i=1}^{\gamma_{\mathcal{S}'}}\alpha'_i\,|\mathrm{GS}'{}_i\rangle \in \operatorname{span}\mathcal{GS}'$. This shows that $\operatorname{span}\mathcal{GS}' \geq \mathcal{PS}'$ and therefore $\operatorname{span}\mathcal{GS}' = \mathcal{PS}'$. That $\gamma_{\mathcal{S}'} = \frac{\gamma_{\mathcal{S}}}{2}$ follows since no new relation between plaquette operators is established due to the B_{new} operator. ■

Applications of the developed calculus are given in section 2.3 below.

2.3 Ground states for small Toric Code systems

Several examples of small Toric Code systems and their corresponding ground states are given in Table 2.1 (see column **Setting** and **GS**). Much more systems are listed in Tables A.2, A.3 and A.4 printed in appendix A. Further explanations regarding the other columns are given there.

[21] See proof of Lemma 2.2.5. Here it is $\mathcal{C}^* = \mathcal{C}' + \mathcal{C}_{pq}$ and consequently $\mathcal{C}^*_{pq} = 0$.

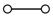
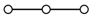
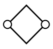
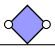
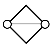
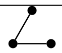
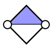
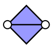
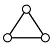
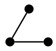
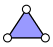
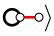
No.	N	**Setting**	γ	**AE**	**NL**	**GS** $\mathcal{N}^{-1}\,\vert\mathrm{GS}_i\rangle$, $1 \leq i \leq \gamma$	**LU/S**	**Comment**
1	1		2^0	no	no	$\vert+\rangle$	●	single qubit
2	2		2^0	no	no	$\vert+\rangle\,\vert+\rangle$	● ●	biseparable
3	2		2^1	no	no	$\vert\Phi^+\rangle,\vert\Psi^+\rangle$	●—●	Bell states
4	2		2^0	(yes)	no	$\vert\Phi^+\rangle$	●—●	Bell state
5	3		2^2	no	no	$\vert\Psi^+\rangle\,\vert+\rangle \pm \vert\Psi^-\rangle\,\vert-\rangle$, $\vert\Phi^+\rangle\,\vert+\rangle \pm \vert\Phi^-\rangle\,\vert-\rangle$		
6	3		2^1	(yes)	no	$\vert\Phi^+\rangle\,\vert+\rangle \pm \vert\Phi^-\rangle\,\vert-\rangle$		
7	3		2^0	(yes)	no	$\vert\Phi^+\rangle\,\vert+\rangle + \vert\Phi^-\rangle\,\vert-\rangle$		GHZ state
8	3		2^1	no	no	$\vert+\rangle^{\otimes 3} \pm \vert-\rangle^{\otimes 3}$		
9	3		2^0	yes	no	$\vert+\rangle^{\otimes 3} + \vert-\rangle^{\otimes 3}$		

TABLE 2.1 (Color online) : TCM spin systems with $1 \leq N \leq 3$ spins. γ is the degeneracy, **AE** states if **A**nyonic **E**xcitations are possible and **NL** if **N**on **L**ocal measurements are necessary to determine topological quantum numbers v_{top}. The column **GS** lists the **G**round **S**tates up to a normalizing constant $\mathcal{N}$. **LU/S** denotes the equivalence class with respect to **L**ocal **U**nitaries and qubit **S**wap operations in a graph state representation.

Example 3: (GHZ STATES). Consider a single TCM plaquette with N spins (for $N = 3$ see system 9 in Table 2.1). The corresponding ground state is a GHZ state in the σ^x basis. To see this, one applies Lemma 2.2.2 and Remark 2.2.6 in the following way:

Since adding a single string leads to product states

$\vert$  $\rangle = \vert$ $\rangle \otimes \vert$ $\rangle$

a $N-1$-qubit string has the (unique) TCM stabilizer state $\vert$ $\rangle^{\otimes N-1} = \vert+\rangle^{\otimes N-1}$. Closing the string with the Nth edge yields the ground state

$\vert$ $\rangle = \frac{1}{\sqrt{2}}\left[\vert\right.$ $\rangle \otimes \vert+\rangle + Z\left[\mathcal{C}_{pq}\right]\vert$ $\left.\rangle \otimes \vert-\rangle\right] = \frac{1}{\sqrt{2}}\left[\vert+\rangle^{\otimes N-1} \otimes \vert+\rangle + Z\left[\mathcal{C}_{pq}\right]\vert+\rangle^{\otimes N-1} \otimes \vert-\rangle\right]$

$$= \frac{1}{\sqrt{2}}\left[\vert+\rangle^{\otimes N} + \vert-\rangle^{\otimes N}\right]$$

where $\mathcal{C}_{pq}$ runs over all $N-1$ qubits. Applying Hadamard gates to each qubit yields $\vert\mathrm{GHZ}_N\rangle$ in the computational σ^z basis.

Example 4: (MULTIPLE GROUND STATES). Here we show how the (two) ground states of system 33 (depicted in Table A.3, appendix A) may be derived from known ground states of smaller systems. First note that system 33 is obtained from system 18 (Table A.2) connecting the outer vertices by a new edge. Again, system 18 may be derived from two copies of system 4 (Table A.1 or 2.1). It is easy to see that system 4 stabilizes a (unique) Bell state $\vert\Phi^+\rangle$. Hence from Lemma 2.2.2 it follows for the ground state of system 18: $\vert\Phi^+\rangle\,\vert\Phi^+\rangle$. Note that $\sigma_1^z\,\vert\Phi^+\rangle = \sigma_2^z\,\vert\Phi^+\rangle = \vert\Phi^-\rangle$. Therefore the (two) ground states, given by Remark 2.2.6, read as follows

$\vert$ $\rangle_{\pm} = \frac{1}{\sqrt{2}}\left[\vert\Phi^+\rangle\,\vert\Phi^+\rangle \otimes \vert+\rangle \pm Z\left[\mathcal{C}_{pq}\right]\vert\Phi^+\rangle\,\vert\Phi^+\rangle \otimes \vert-\rangle\right]$

$$= \frac{1}{\sqrt{2}}\left[\vert\Phi^+\rangle\,\vert\Phi^+\rangle\,\vert+\rangle \pm \vert\Phi^-\rangle\,\vert\Phi^-\rangle\,\vert-\rangle\right]$$

Here the topological quantum number is $v_{\pm} = \pm 1$ and is defined by a z-type loop operator winding once around the "hole". The path $\mathcal{C}_{pq}$ runs from the left vertex of system 18 to the right one. Obviously there are four different possibilities to choose $\mathcal{C}_{pq}$. However, since $\sigma_1^z\,\vert\Phi^+\rangle = \sigma_2^z\,\vert\Phi^+\rangle = \vert\Phi^-\rangle$ it does not matter which one is chosen.

Chapter 3

LU/SWAP classification of TCM ground states

3.1 Objective and overview

To reach our goal – classifying Toric Code ground states with respect to local unitary operations and the permutation of qubits, that is – we are going to apply results published in refs. [1,2,5] in a straightforward way.

The approach is as follows:

1. According to the theory of TCMs different ground states are transformed into each other by application of string operators. These operators (see sec. 1.3) are *local* in the sense of LU-equivalence. Therefore we do not have to deal with different ground states and consequently may reduce the considered systems to the non-degenerate ones (i.e. the topological quantum numbers of all loops are stabilized to +1 by adding appropriate plaquettes).

2. The binary algorithm developed by Van den Nest *et al.* [5] is used to transform the TCM stabilizer into a graph state stabilizer. This stabilizer describes a LC-equivalent graph state of the Toric Code system under consideration.

3. By graph state transformations using local Clifford operations one converts the graph state obtained in the binary framework to one of the LU/SWAP-class representatives computed in ref. [2]. This allows the classification of TCM ground states with up to 7 spins by means of simple algebra.

Fig. 3.1 illustrates the relations described above. This algorithm is described together with some examples in the subsequent section.

$$\begin{tikzcd}
\{|v_1,\dots,v_n\rangle_A\}_{v_i\in\{1,-1\}} \arrow[r, leftrightarrow, "\text{LC}", "\text{String operators}"'] \arrow[d, leftrightarrow, "\text{LC/SWAP?}"] & |1,\dots,1\rangle_A \arrow[r, leftrightarrow, "\text{LC}", "\text{Hadamard gates}"'] & |G\rangle_A \arrow[r, leftrightarrow, "\text{LC/SWAP}", "\text{LC-rule}"'] & |G^*\rangle_A \arrow[d, leftrightarrow, "\text{LC/SWAP!}"] \\
\{|v_1,\dots,v_n\rangle_B\}_{v_i\in\{1,-1\}} \arrow[r, leftrightarrow, "\text{LC}", "\text{String operators}"'] & |1,\dots,1\rangle_B \arrow[r, leftrightarrow, "\text{LC}", "\text{Hadamard gates}"'] & |G\rangle_B \arrow[r, leftrightarrow, "\text{LC/SWAP}", "\text{LC-rule}"'] & |G^*\rangle_B
\end{tikzcd}$$

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{Kitaev [1]}}\;\underbrace{\qquad\quad}_{\text{Van den Nest } et\ al.\ [5]}\;\underbrace{\qquad\qquad\qquad\qquad}_{\text{Hein } et\ al.\ [2]}$$

FIGURE 3.1 (Color online) : Symbolic illustration of the equivalences used in the algorithm to classify TCM ground states. Here **LC** denotes *local Clifford operations* (hence local unitaries) and **SWAP** denotes arbitrary permutations of qubits.

The second part of this chapter is dedicated to a purely graph theoretic transformation to compute LC-equivalent graph states from a given TCM stabilizer defined by a graph. First, we are going to motivate the graph theoretic approach by referring to the examples used in the first part. Secondly, the conjecture is proved and finally this rule is applied to some small Toric Code systems.

3.2 Mapping TCM ground states to graph states

3.2.1 Algorithmic approach

Let $\mathcal{S}$ be a stabilizer with (independent) set of generators $\mathcal{G}$ and $\mathbf{M} \equiv \mathbf{M}[\mathcal{G}]$ its binary representation. We are going to apply local unitary operations (more precisely: local Clifford operations) on the stabilizer state by transforming the Pauli operators in the generator accordingly. Subsequently a basis change is performed to show that the new stabilizer state is indeed a graph state.

The detailed algorithm as derived in [5] reads as follows:

1. The first step involves a basis change $\mathbf{T}' \equiv \mathbf{T}[\mathcal{G},\mathcal{G}'] \in \mathrm{GL}(N,\mathbb{F}_2)$ such that the generator matrix is transformed in the following way:

$$\mathbf{M} = \begin{bmatrix} \mathbf{Z} \\ \mathbf{X} \end{bmatrix} \quad \rightarrow \quad \mathbf{M}' = \mathbf{M}\mathbf{T}' = \begin{bmatrix} \mathbf{R}_z & \mathbf{S}_z \\ \mathbf{R}_x & \mathbf{0} \end{bmatrix} \tag{3.1}$$

   where $\mathbf{R}_x$ is a $N \times K$-Matrix with full rank, i.e. $\operatorname{rank}\mathbf{R}_x = K = \operatorname{rank}\mathbf{X} \leq N$. Note that if one sorts the TCM stabilizer generators appropriately (independent A_s operators on the left and B_p operators on the right) then a TCM stabilizer has already this structure and this first step is unnecessary.

2. Since $\mathbf{R}_x$ has rank K one can obtain the following form by permuting the rows of the matrix

$$\mathbf{R}_x = \begin{bmatrix} \mathbf{R}_x^1 \\ \mathbf{R}_x^2 \end{bmatrix} \tag{3.2}$$

   such that $\mathbf{R}_x^1$ is an invertible $K \times K$-submatrix of $\mathbf{R}_x$. Since the order of the rows in the generator matrix corresponds to the labeling of the qubits this transformation represents just a relabeling of the qubits in question.

   In case of a TCM stabilizer one can label the spins appropriately using the graph theoretic approach described in the next section such that $\mathbf{R}_x$ has this structure from the beginning and this transformation can be omitted, too.

3. Divide $\mathbf{S}_z$ and $\mathbf{R}_z$ according to the structure found for $\mathbf{R}_x$, i.e.

$$\mathbf{S}_z = \begin{bmatrix} \mathbf{S}_z^1 \\ \mathbf{S}_z^2 \end{bmatrix} \quad \text{and} \quad \mathbf{R}_z = \begin{bmatrix} \mathbf{R}_z^1 \\ \mathbf{R}_z^2 \end{bmatrix} \tag{3.3}$$

   such that $\mathbf{S}_z^1$ is a $K \times (N-K)$ matrix and $\mathbf{S}_z^2$ has the dimension $(N-K) \times (N-K)$ (the same holds for $\mathbf{R}_z^1$ and $\mathbf{R}_z^2$). One can show [5] that the self-orthogonality of the stabilizer with respect to the symplectic product[22] implies that $\mathbf{S}_z^2$ is invertible.

   This step actually involves no transformation at all.

4. The following step is the only one where the considered stabilizer state is *transformed* by local Clifford operations. In particular, Hadamard gates H_i acting on the qubits with indices $K+1,\ldots,N$ require the following transformation of 1-qubit Pauli operators in the stabilizer:

$$\begin{aligned} \sigma^x &\rightarrow \sigma^z \\ \sigma^z &\rightarrow \sigma^x \end{aligned}$$

[22] Formally this reads $\mathbf{M}^T\mathbf{P}\mathbf{M} = \mathbf{0}$.

The exchange of x- and z-type Pauli operators acting on the qubits represented by the last $N-K$ rows in the upper and lower block of the generator matrix $\mathbf{M'}$ can be achieved in the binary picture by swapping the last $N-K$ rows of the upper block with the corresponding rows of the lower one. This yields the form

$$\mathbf{M}'' = \mathbf{HMT}' = \begin{bmatrix} \mathbf{R}_z^1 & \mathbf{S}_z^1 \\ \mathbf{R}_x^2 & \mathbf{0} \\ \mathbf{R}_x^1 & \mathbf{0} \\ \mathbf{R}_z^2 & \mathbf{S}_z^2 \end{bmatrix} \quad \Rightarrow \quad \mathbf{X}'' := \begin{bmatrix} \mathbf{R}_x^1 & \mathbf{0} \\ \mathbf{R}_z^2 & \mathbf{S}_z^2 \end{bmatrix} \in \mathrm{GL}\left(N, \mathbb{F}_2\right) \tag{3.4}$$

where $\mathbf{H} \in \mathcal{C}_N^l$ denotes the $N-K$ Hadamard transformations as a local Clifford operator. Since both $\mathbf{R}_x^1$ and $\mathbf{S}_z^2$ are invertible the lower $N \times N$-block $\mathbf{X}''$ of the new generator matrix has full rank.

Note that in the special case of TCM stabilizers it follows $\mathbf{R}_z = \mathbf{0}$ since A_s operators are constructed from σ^x Pauli operators exclusively.

5. Let $\mathbf{T}'' \equiv \left(\mathbf{X}''\right)^{-1} \in \mathrm{GL}\left(N, \mathbb{F}_2\right)$ be the inverse matrix of $\mathbf{X}''$. Multiplication from the right – representing a basis change in the stabilizer framework – yields

$$\mathbf{M}''' = \mathbf{HMT}'\mathbf{T}'' = \begin{bmatrix} \mathbf{\Gamma} \\ \mathbf{E}_N \end{bmatrix} \tag{3.5}$$

Since this is a binary representation of a stabilizer the self-orthogonality relation $\left(\mathbf{M}'''\right)^T \mathbf{P} \left(\mathbf{M}'''\right) = \mathbf{0}$ holds. It is easy to show that this implies the symmetry of $\mathbf{\Gamma}$, i.e. $\mathbf{\Gamma}^T = \mathbf{\Gamma}$.

To set $\Gamma_{i,i} = 0$ for all $i \in \{1, \ldots, N\}$ one applies further LC-operations on the concerned spins [5]. In the case of TCM stabilizers it turns out that this is not necessary, however.

In the end the generator matrix $\mathbf{M}'''$ has the *graph state form* defined in 1.2.6. $\mathbf{\Gamma}$ is the adjacency matrix of the LC-equivalent graph state.

This algorithm seems to be a little cumbersome in application but, as we will see in the following examples, in the case of TCM stabilizers it is quite easy to follow since some of the steps described above are not necessary if one chooses the labeling appropriately.

Example 5: (Algorithmic approach). Consider the Toric Code system depicted twice in fig. 3.2, each with different qubit labels. We are going to use the algorithm described above to transform both systems into their LC-equivalent graph state which is depicted on the right-hand side of each Toric Code system.

The reader should recall the difference between the geometric description of both stabilizers: The vertices of a TCM graph (white and green dots) represent A_s stabilizer operators and the edges spins whereas the vertices of a graph defining a graph state (black dots) represent spins and the edges define stabilizer operators connecting these spins.

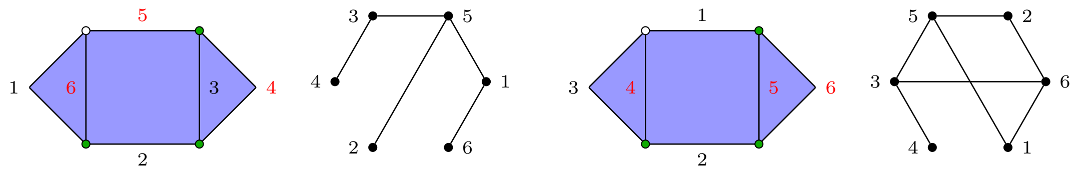


Figure 3.2 (Color online) : A TCM stabilizer with different spin labelings. The LC-equivalent graph states computed in the binary framework are depicted on the right-hand side of each TCM stabilizer. The red numbers mark spins a Hadamard transformation is applied to. The green vertices represent an *independent* set of star operators.

Let us compute both graph states step by step. Note that the rows of the generator matrices are mapped to the spins in increasing order according to their labels. The Hadamard transformations are applied to the spins marked red in fig. 3.2. The choice of the red marked spins is not arbitrary as we will see in the next section.

(i) Consider the TCM stabilizer in fig. 3.2 *on the left.* According to the spin numbering we can read off the generator matrix with star operators on the left (first 3 columns) and plaquette operators on the right (last 3 columns). Note that we may choose three of the four A_s operators defined by the graph arbitrarily since only three independent star operators exist. The same argument holds for the four plaquettes (the fourth plaquette fills the *outside* of the graph). If we choose the three blue plaquettes and the three green star operators, the generator matrix reads as follows

$$\mathbf{M}_1 = \begin{bmatrix} 0&0&0&1&0&0\\ 0&0&0&0&1&0\\ 0&0&0&0&1&1\\ 0&0&0&0&0&1\\ 0&0&0&0&1&0\\ 0&0&0&1&1&0\\ 1&0&0&0&0&0\\ 1&1&0&0&0&0\\ 0&1&1&0&0&0\\ 0&1&1&0&0&0\\ 0&0&1&0&0&0\\ 1&0&0&0&0&0 \end{bmatrix} \quad \text{where} \quad \mathbf{R}_x = \begin{bmatrix} 1&0&0\\ 1&1&0\\ 0&1&1\\ 0&1&1\\ 0&0&1\\ 1&0&0 \end{bmatrix} \quad \text{and} \quad \mathbf{R}_x^1 = \begin{bmatrix} 1&0&0\\ 1&1&0\\ 0&1&1 \end{bmatrix} \in \mathrm{GL}\left(3, \mathbb{F}_2\right)$$

Therefore the first three steps in the algorithm can be skipped. Thus write $\mathbf{M}_1' = \mathbf{M}_1$ and apply Hadamard transformations to the spins 4, 5 and 6. This yields

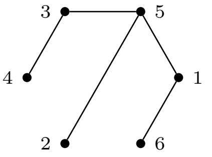
$$\mathbf{M}_1'' = \mathbf{H}\mathbf{M}_1' = \begin{bmatrix} 0&0&0&1&0&0\\ 0&0&0&0&1&0\\ 0&0&0&0&1&1\\ 0&1&1&0&0&0\\ 0&0&1&0&0&0\\ 1&0&0&0&0&0\\ 1&0&0&0&0&0\\ 1&1&0&0&0&0\\ 0&1&1&0&0&0\\ 0&0&0&0&0&1\\ 0&0&0&0&1&0\\ 0&0&0&1&1&0 \end{bmatrix} \quad \text{and} \quad \mathbf{X}'' = \begin{bmatrix} 1&0&0&0&0&0\\ 1&1&0&0&0&0\\ 0&1&1&0&0&0\\ 0&0&0&0&0&1\\ 0&0&0&0&1&0\\ 0&0&0&1&1&0 \end{bmatrix} \in \mathrm{GL}\left(6, \mathbb{F}_2\right)$$

Since this is a generator matrix with invertible lower block one can proceed with step five and apply the inverse $\mathbf{T}'' = \left(\mathbf{X}''\right)^{-1}$ to obtain the graph state form of the stabilizer:

$$\mathbf{M}_1''' = \mathbf{M}_1''\mathbf{T}'' = \begin{bmatrix} 0&0&0&1&0&0\\ 0&0&0&0&1&0\\ 0&0&0&0&1&1\\ 0&1&1&0&0&0\\ 0&0&1&0&0&0\\ 1&0&0&0&0&0\\ 1&0&0&0&0&0\\ 1&1&0&0&0&0\\ 0&1&1&0&0&0\\ 0&0&0&0&0&1\\ 0&0&0&0&1&0\\ 0&0&0&1&1&0 \end{bmatrix} \cdot \begin{bmatrix} 1&0&0&0&0&0\\ 1&1&0&0&0&0\\ 1&1&1&0&0&0\\ 0&0&0&0&1&1\\ 0&0&0&0&1&0\\ 0&0&0&1&0&0 \end{bmatrix} = \begin{bmatrix} 0&0&0&0&1&1\\ 0&0&0&0&1&0\\ 0&0&0&1&1&0\\ 0&0&1&0&0&0\\ 1&1&1&0&0&0\\ 1&0&0&0&0&0\\ 1&0&0&0&0&0\\ 0&1&0&0&0&0\\ 0&0&1&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&0\\ 0&0&0&0&0&1 \end{bmatrix}$$

Thus we get the adjacency matrix and the corresponding graph as a representation of the graph state stabilizer:

$$\mathbf{\Gamma}_1 = \begin{bmatrix} 0&0&0&0&1&1\\ 0&0&0&0&1&0\\ 0&0&0&1&1&0\\ 0&0&1&0&0&0\\ 1&1&1&0&0&0\\ 1&0&0&0&0&0 \end{bmatrix} \longleftrightarrow$$



Therefore we found a LC-equivalent graph state representation of the considered Toric Code system. More precisely, we obtained a LC-equivalent representation of *all* TCM ground states that may be present in a TCM system defined by the graph (i.e. the A_s operators) only, disregarding the (blue) plaquettes. This holds since plaquette operators only *restrict* the set of allowed ground states and these ground states can be transformed into each other by application of string operators, local Clifford operations, that is.

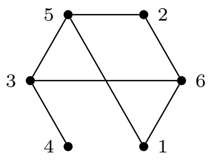
(ii) Consider the TCM stabilizer in fig. 3.2 *on the right*. The spins were relabeled since spins with Hadamard transformations applied to have to be encoded in the last rows of the generator matrix. Hence we took account for step two by relabeling the spins.

Once again, we choose the blue plaquette and the green star operators to establish the binary representation of the TCM stabilizer. This yields

$$\mathbf{M}_2' = \mathbf{M}_2 = \begin{bmatrix} 0&0&0&0&1&0\\ 0&0&0&0&1&0\\ 0&0&0&1&0&0\\ 0&0&0&1&1&0\\ 0&0&0&0&1&1\\ 0&0&0&0&0&1\\ 0&0&1&0&0&0\\ 1&1&0&0&0&0\\ 1&0&0&0&0&0\\ 1&0&0&0&0&0\\ 0&1&1&0&0&0\\ 0&1&1&0&0&0 \end{bmatrix} \quad \Rightarrow \quad \mathbf{M}_2'' = \mathbf{H}\mathbf{M}_2' = \begin{bmatrix} 0&0&0&0&1&0\\ 0&0&0&0&1&0\\ 0&0&0&1&0&0\\ 1&0&0&0&0&0\\ 0&1&1&0&0&0\\ 0&1&1&0&0&0\\ 0&0&1&0&0&0\\ 1&1&0&0&0&0\\ 1&0&0&0&0&0\\ 0&0&0&1&1&0\\ 0&0&0&0&1&1\\ 0&0&0&0&0&1 \end{bmatrix}$$

In the last step the basis transformation $\mathbf{T}''$ is applied ($\mathbf{M}_2''' = \mathbf{M}_2''\mathbf{T}''$) which results in the following stabilizer representation:

$$\mathbf{M}_2''' = \begin{bmatrix} 0&0&0&0&1&1\\ 0&0&0&0&1&1\\ 0&0&0&1&1&1\\ 0&0&1&0&0&0\\ 1&1&1&0&0&0\\ 1&1&1&0&0&0\\ 1&0&0&0&0&0\\ 0&1&0&0&0&0\\ 0&0&1&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&0\\ 0&0&0&0&0&1 \end{bmatrix} \quad \Rightarrow \quad \mathbf{\Gamma}_2 = \begin{bmatrix} 0&0&0&0&1&1\\ 0&0&0&0&1&1\\ 0&0&0&1&1&1\\ 0&0&1&0&0&0\\ 1&1&1&0&0&0\\ 1&1&1&0&0&0 \end{bmatrix} \longleftrightarrow$$



Obviously both graphs obtained as a LC-equivalent representation of one and the same TCM stabilizer state are not equivalent under graph isomorphisms (the first is a tree, the second is not). In fact, there are two differences interwoven: First, there was a relabeling of spins. Therefore we cannot expect to obtain the *same* graph. But relabeling cannot lead to non-isomorphic graphs since graph isomorphisms describe such permutations of vertices. The reason that we obtained a new *topology* of the representing graph is due to the different choice of spins that were transformed by Hadamard gates. These two sets of spins *cut* the TCM graph in different ways, thus leading to different (nevertheless LC-equivalent) graph state representatives.

That these two representatives belong to the same LU/SWAP-class is shown by the LC-rule (see Proposition 1.6.3). We obtain the series of LC-equivalent graph states (or LC-equivalent graphs) depicted in fig. 3.3.

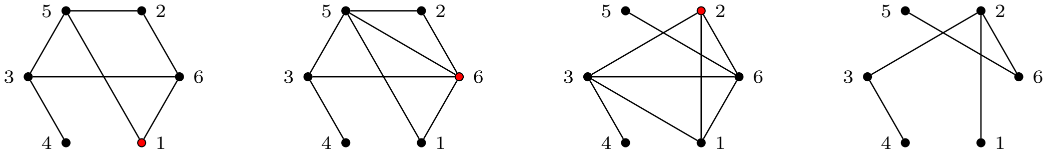


Figure 3.3 (Color online) : Transformation of one graph state into another under local Clifford operations. Here the LC-rule (see subsec. 1.6.2 and Example 2) was used to show that both graph state representatives belong to the same LU/SWAP-class.

The result of three successive local complementations is a graph that is isomorphic to the graph obtained as result of the first part. This is illustrated in fig. 3.4 where the graph is redrawn and subsequently a permutation on the vertex set is applied.

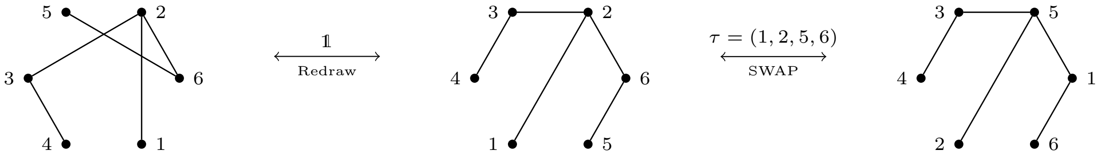


Figure 3.4 (Color online) : The vertex permutation (relabeling) resembles several qubit SWAP operations.

Therefore we showed that both graph state representatives are equivalent under local unitary operations (more precisely: local Clifford operations) and permutations of spins (SWAP operations). The SWAP operations were necessary since we already started with a spin-permuted system. It is important to note that these qubit permutations have consequences for the physical implementation of such systems *if they are not interpreted as a pure relabeling of qubits but as a physical exchange of particles.* Imagine we prepared a system of six qubits (e.g. as Rydberg states or trapped ions). We may label the atoms/ions representing a qubit with $1, 2, \dots, 6$. If these qubits are prepared in the TCM ground state defined by the left Toric Code system in fig. 3.2 (according to the given edge labels), then we showed above that this state *cannot* be transformed into the ground state of the right Toric Code system in fig. 3.2 by means of *local unitaries alone*[23]. This is due to the fact that additional qubit SWAP operations are necessary to transform one of these states into the other. But a qubit SWAP gate is not realisable by local unitaries in general.

Remark 3.2.1. *The reader may notice that the graphs obtained are bipartite (or 2-colorable). This is the case in general since the upper and lower blocks of the generator matrix are diagonal or anti-diagonal block matrices after the Hadamard transformation. This yields an anti-diagonal adjacency block matrix which represents a bipartite graph. In fact, this is a consequence of the special structure of the stabilizer generators.*

3.2.2 Graph theoretic approach

Transforming TCM ground states (or their stabilizers) into LC-equivalent graph states is feasible with the algorithm explained above. But in most cases one needs a CAS to invert and multiply the (quite large) matrices if one does not want to compute them by hand.

[23] In actual fact one has to show that both graphs belong to *different* LC-classes. Using the techniques developed in secs. 4.2 and B.1 this proves true. LC $\Leftrightarrow$ LU holds since $N = 6 \leq 7$.

Therefore it begs the questions: Is there a geometrical way of transforming a given TCM stabilizer into LC-equivalent graph states? And if so, how does it look like?

In fact, it turns out that there is such an algorithm that may be stated in purely graph theoretic terms. This leads us to the central statement of this chapter:

Theorem 3.2.2: (Graph transformation rule). *Let $\mathcal{L}$ be the graph/lattice of a generalised Toric Code Model as defined in 1.5.1. Then there is a graph state $|\mathcal{X}\rangle \in \mathcal{H}^N$ to which all basis vectors $|\mathbf{v}\rangle$ of the protected space $\mathcal{PS}$ are equivalent under local Clifford operations. A graph $\mathcal{X}$ defining such a graph state can be constructed from $\mathcal{L}$ in the following way:*

1. *Find an arbitrary* ***spanning tree*** *$\mathcal{L}'$ of $\mathcal{L}$ by deleting a set $\mathcal{E} \subseteq E(\mathcal{L})$ which has the properties*

   i. *that there is no cycle $\mathcal{C}$ on $\mathcal{L}$ such that $\mathcal{C} \cap \mathcal{E} = \emptyset$*

   ii. *and $\mathcal{E}$ is irreducible in the sense that removing an arbitrary edge from $\mathcal{E}$ violates the first property.*

   Then set $\mathcal{L}' := \mathcal{L} - \mathcal{E}$.

2. *Now let $V(\mathcal{X}) := E(\mathcal{L})$ be the vertex set of a new graph $\mathcal{X}$. This graph is defined by his edge set $E(\mathcal{X})$ as follows: Let $r, s \in V(\mathcal{X})$. Then $\{r, s\}$ is an edge of $\mathcal{X}$ if and only if $\{r, s\} \cap \mathcal{E} \neq \emptyset$ (w.l.o.g. let r be in $\mathcal{E}$) and if there is a path $\mathcal{C}_{pq}$ from p to q on $\mathcal{L}'$ $(p, q \in V(\mathcal{L}') = V(\mathcal{L}))$ such that $r = \{p, q\}$ and $s \in \mathcal{C}_{pq}$.*

Before proving this statement let us consider the example used for the algorithm in the binary framework and show how Theorem 3.2.2 allows us to derive the corresponding graphs without any calculation. To this end we state the following

Remark 3.2.3. *The choice of a spanning tree $\mathcal{L}'$ is arbitrary. Changing this choice may or may not lead to non-isomorphic graphs as graph state representations. Thus the graph resulting from the algorithm described in Theorem 3.2.2 is not unique. Nevertheless – as the proof confirms – the graphs obtained are equivalent under local complementations (see sec. 3.4).*

Example 6: (Graph theoretic approach). Consider once again the Toric Code system depicted twice in fig. 3.2. In fig. 3.5 this original system is shown on top. Choose two different spanning trees by deleting three edges in two different ways. The two spanning trees we are going to use are shown in the second row of fig. 3.5. The reader may notice that we deleted exactly the edges which are labeled red in fig. 3.2, i.e. the spins transformed by Hadamard operations. The connection between both algorithms responsible for this phenomenon[24] will become clear in the proof of Theorem 3.2.2.

Since we found spanning trees of the TCM stabilizer in question one can construct the LC-equivalent graph states by applying Theorem 3.2.2. The approach is as follows:

Consider a deleted edge (drawn dashed). The spin located at this edge (yellow) will be a vertex of the new graph. Next, find the path in the spanning tree (there is one and only one, see proof below) starting at one vertex of the TCM lattice adjacent to the deleted edge and ending at its other adjacent vertex. In the last step, connect the spin on the deleted edge with all spins (green) on the path we just found.

This procedure is performed for all spins on deleted edges (in our example there are three of them). We get the red drawn graphs depicted in the third row of fig. 3.5. These graphs represent the LC-equivalent graph states we were looking for and are easily seen to be isomorphic to the graphs depicted in fig. 3.2 and derived in the binary framework. As we showed earlier (see figs. 3.3 and 3.4) both graphs are equivalent under local complementations (and qubit permutations) and therefore describe LC/SWAP-equivalent graph states.

The reader may have noticed that by this construction a yellow vertex cannot be connected to another vertex of this kind. Analogously green vertices may only be connected to yellow vertices.

[24] In fact, the reason for choosing the red marked spins for Hadamard operations in the previous section was that they "cut" the TCM graph and thus create a spanning tree.

Consequently the graph obtained is *bipartite* or 2-*colorable*, meaning that there are two sets of vertices only connected to vertices of the other set. Consequently such graphs can only feature cycles with an even number of edges.

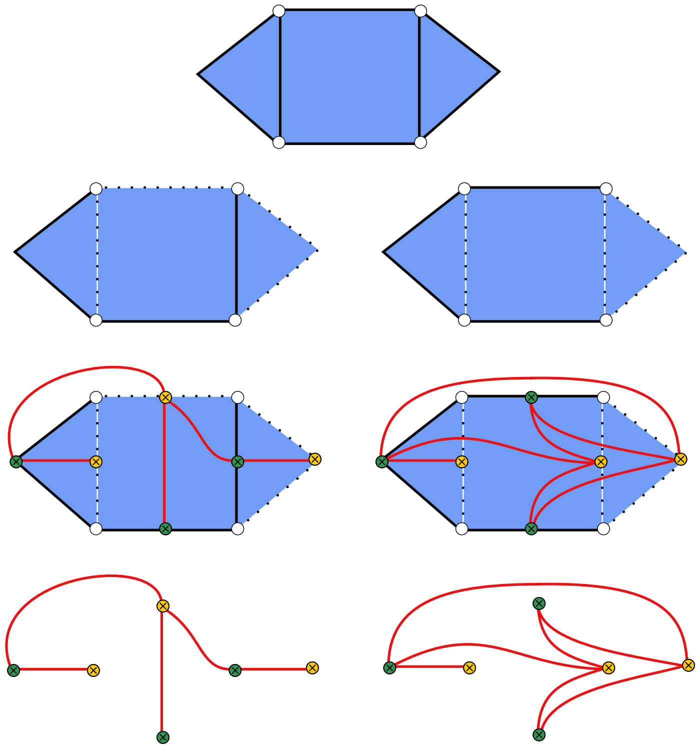


Figure 3.5 (Color online) : Transformation of a TCM stabilizer into LC-equivalent graph states using the graph theoretic approach stated in Theorem 3.2.2. Here we choose two different spanning trees (second row) by deleting three edges (dashed lines) such that no cycle of the original lattice survives. Connecting the spins on the deleted edges with all spins on the spanning tree path with endpoints adjacent to the deleted edge yields the red graphs (third row). These graphs (fourth row) are easily seen to be isomorphic to the graphs obtained by computations in the binary framework.

The last part of this section is dedicated to the proof of Theorem 3.2.2.

Proof. That for a non-degenerate Toric Code stabilizer (i.e. every loop of the lattice is covered by a plaquette) there exist LC-equivalent graph states follows from the general equivalence of stabilizer codes and graph states (see Proposition 1.7.2). That the basis states of the protected space as defined in the introduction are LC-equivalent follows from the fact that they may be transformed into each other by string operators, which are products of Pauli operators. From the viewpoint of LU-classification this leads to simplifications. Without loss of generality the considered TCM stabilizer has the following properties:

- The graph defined by the star operators is a multigraph.
- For each cycle in this multigraph there is a z-type string operator in the stabilizer.

In the following it is derived how the binary transformation can be expressed in the geometric picture.

Consider a lattice $\mathcal{L}$ describing a TCM stabilizer $\mathcal{S}$ with independent generators $A_s, B_p \in \mathcal{G}$. Identify this lattice (ignoring the plaquettes) with the multigraph defined by the star operators A_s. Let $\mathcal{L}'$ be an arbitrary spanning tree and $\mathcal{E}$ the set of edges such that $\mathcal{L}' = \mathcal{L} - \mathcal{E}$. In fig. 3.6 this is depicted in A and B where the dashed lines represent $\mathcal{E}$. Apply Hadamard transformations H to the spins in $\mathcal{E}$ described by the (unique) stabilizer state $|\mathbf{v}\rangle$ of $\mathcal{S}$. In the stabilizer picture this transformation is represented by the substitution

$$\begin{aligned} \sigma^x &\rightarrow \sigma^z \\ \sigma^z &\rightarrow \sigma^x \end{aligned}$$

Consider one of these transformed qubits and the edge $e = \{p, q\} \in E(\mathcal{L})$ it belongs to ($p, q \in V(\mathcal{L})$). The adjacent vertices p and q belong to both $\mathcal{L}$ and $\mathcal{L}'$. Then there exists one and only one path $\mathcal{C}_{pq}$ on $\mathcal{L}'$ connecting p and q. This is shown in fig. 3.6 C for the yellow spin. That there exists one follows since $\mathcal{L}'$ is connected by definition (and $\mathcal{L}$ is connected by assumption). Given two different paths $\mathcal{C}^1_{pq}$ and $\mathcal{C}^2_{pq}$ yields a non trivial closed path $\mathcal{C}^1_{pq} + \mathcal{C}^2_{pq}$ on $\mathcal{L}'$. But this is not possible since $\mathcal{L}'$ is a tree. Therefore $\mathcal{C}_{pq}$ is unique. Let $\sigma^z_e \otimes Z[\mathcal{C}_{pq}]$ be a combination of plaquette operators in $\mathcal{S}$ (this is possible since $Z[\mathcal{C}] \in \mathcal{S}$ for a closed path $\mathcal{C}$ as mentioned above). This stabilizer operator is transformed under the Hadamard transformation as follows:

$$\sigma^z_e \otimes Z[\mathcal{C}_{pq}] = \sigma^z_e \otimes \bigotimes_{i \in \mathcal{C}_{pq}} \sigma^z_i \quad \longrightarrow \quad K^{(e)}_{\mathcal{X}} = H_e \sigma^z_e \otimes Z[\mathcal{C}_{pq}] H^\dagger_e = \sigma^x_e \otimes \bigotimes_{i \in \mathcal{C}_{pq}} \sigma^z_i \tag{3.6}$$

Obviously this is a graph state generator of the graph $\mathcal{X}$ as defined in Theorem 3.2.2 since in this graph the vertex e is connected to all edges in $\mathcal{L}$ that belong to $\mathcal{C}_{pq}$ (see fig. 3.6 D). This argument holds for every (deleted) edge in $\mathcal{E}$. Therefore we find that the transformed stabilizer $\mathcal{S}' := H\mathcal{S}H^\dagger$ comprises at least the graph state stabilizer generators of $\mathcal{X}$ for the vertices that can be identified with edges in $\mathcal{E}$.

Note that at this stage all physical transformations are already done. What follows is a purely mathematical transformation of the stabilizer to show that its stabilized state is a graph state[25]. Therefore it remains to show that the graph state generators for the vertices identified with edges of the spanning tree (marked green in fig. 3.6 D) are in $\mathcal{S}'$.

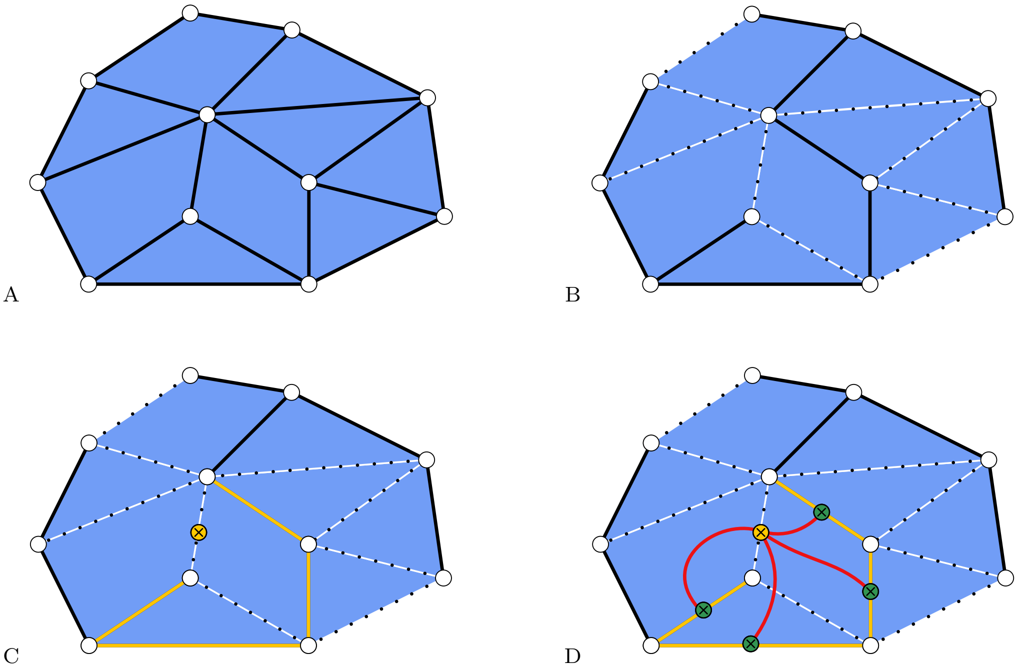


Figure 3.6 (Color online) : The three essential steps to transform a graphical representation of a TCM stabilizer into a LC-equivalent graph state. A: The given TCM stabilizer. B: Choose a spanning tree (black) by deleting (dashed) edges. C: For each spin on a deleted edge find the path on the tree connecting the adjacent vertices. D: Connect each spin on a deleted edge with all spins on his corresponding path.

To this end consider an arbitrary spin on the spanning tree. In fig. 3.7 this spin is marked green. The corresponding edge of the spanning tree cuts the latter into two parts since there is no loop in a tree. This construction induces a bipartition of the vertex set $V(\mathcal{L}') = V_r \dot{\cup} V_b$ (in the figure the red vertices belong to V_r and the blue to V_b). Due to this partition two different types of deleted edges in $\mathcal{E}$ are defined: The one type connects vertices of the same partition (continuous

[25] In the binary framework this is done by the basis transformation in step five.

white lines) and the other type (dashed lines) interconnects both sets of vertices V_b and V_r. It is now easy to see that a spin on a deleted edge (yellow) is connected to the (green) spin iff it belongs to a deleted edge interconnecting both subsets for a path $\mathcal{C}_{pq}$ inevitably hits the (green) spin under consideration.

Therefore we characterised all spins on deleted edges that created an edge to the spin on the spanning tree due to the previous edge-construction step. With this construction in mind we return to the transformed stabilizer $\mathcal{S}'$.

In fact, the following combination of operators in $\mathcal{S}'$ yields the graph state generator for a spin $e' \in E(\mathcal{L}')$ on the spanning tree

$$K_{\mathcal{X}}^{(e')} = \prod_{s \in V_r} A'_s \quad \text{where} \quad A'_s = H A_s H^{\dagger}$$

Here $H \in \mathcal{C}_N^l$ is the combination of (local) Hadamard transformations applied to the TCM ground state. Note that V_r may be replaced by V_b in the equation above. That this formula holds true can be seen in the following way: By applying a subset of transformed A'_s operators a spin can be affected in three different ways.

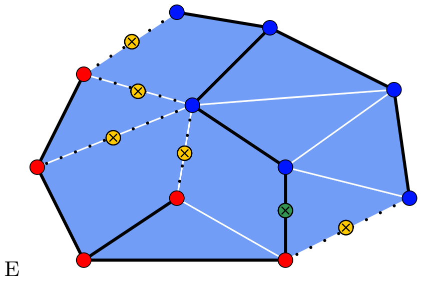


FIGURE 3.7 (Color online) : Bipartition of the spanning tree vertex set (red and blue vertices) used to construct the graph state stabilizer generator for the green spin.

First, if it belongs to an edge with adjacent vertices of the same partition nothing happens since either there are two operators acting simultaneously (thus acting trivially on the considered spin) or no operation is performed at all. Note that this holds true even for the transformed “star operators” A'_s since $\sigma^z\sigma^z = \mathbb{1}$. The second case occurs if the adjacent vertices belong to different subsets and the spin under consideration is *not* e' (yellow in fig. 3.7). Then a non-trivial operation is performed on this spin. This case can only apply to spins on deleted edges, hence the operation is described by σ^z. The last case equals the previous one with the only difference that the spin under consideration *is* e'. As in the previous case a non trivial operation is performed, but now it is described by σ^x.

Summing up, we found that $K_{\mathcal{X}}^{(e')}$ acts as σ^x on e', as σ^z on all vertices f on deleted edges such that $\{e', f\} \in E(\mathcal{X})$ and trivially otherwise. Hence it is the graph state stabilizer generator we were looking for. Therefore we showed that $\mathcal{S}[\mathcal{X}] \leq \mathcal{S}'$ where $\mathcal{S}[\mathcal{X}]$ denotes the graph state stabilizer defined by $\mathcal{X}$. To show $\mathcal{S}[\mathcal{X}] = \mathcal{S}'$ we note that $\operatorname{rank}\mathcal{S}[\mathcal{X}] = N = \operatorname{rank}\mathcal{S} = \operatorname{rank}\mathcal{S}'$.

This concludes the proof.

■

Remark 3.2.4. *To provide a link to graph theory we mention that the support of the string operator $\sigma_e^z \otimes Z[\mathcal{C}_{pq}]$ (i.e. the closed path $\mathcal{C}_{pq} + e$) is known as the* ***fundamental cycle*** *$\mathcal{C}_e$ of e with respect to $\mathcal{L}'$ and the set of deleted edges connecting V_b and V_r together with e' is called* ***fundamental cut*** *$D_{e'}$ of $\mathcal{L}'$.*

3.3 Examples

Several examples of LC-equivalent graph states for small Toric Code systems can be found in Tables A.1, A.2, A.3 and A.4 (see appendix A). These graphs were derived by means of both methods, the binary framework algorithm described in subsection 3.2.1 and the graph theoretic rule derived in subsection 3.2.2.

The following two examples are dedicated to connections between the graph theoretic rule and the calculus derived in section 2.2.

Example 7: (PRODUCT STATES). According to Lemma 2.2.2 the ground states of two Toric Code systems that are connected by only one common vertex are product states, i.e.

$$|\infty\rangle = |\circ\rangle \otimes |\circ\rangle$$

To this end consider the setting depicted in fig. 3.8.
If one chooses a spanning tree and constructs a LC-equivalent graph state it turns out to be disconnected. Two disconnected graphs denote the tensor product of graph states [8]. Thus we confirmed Lemma 2.2.2 in a purely geometric way since local unitaries cannot transform product states into entangled states and vice versa.
That the graph states constructed from one-vertex connected Toric Code systems (as depicted in fig. 3.8) are disconnected in general is easy to show:

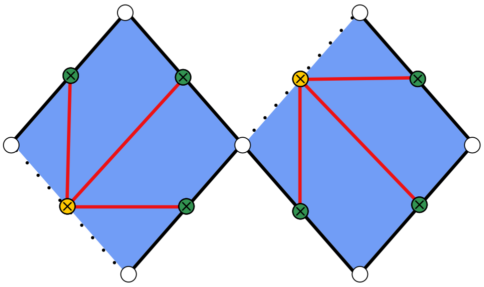


FIGURE 3.8 (Color online) : A Toric Code system with separable ground state. Every spanning tree leads to two disconnected graphs denoting the product of graph states.

The only possibility to connect a spin of one component to components connected by a single vertex would be a fundamental cycle starting at an adjacent vertex of the considered spin, crossing the vertex connecting both components, finally returning to the other adjacent vertex of the spin. Since this cycle must pass the vertex connecting the components *twice* it forms a closed path on one component. This is not possible on a tree, hence it follows that vertices on two one-vertex connected components cannot be connected in the corresponding graph states.

Example 8: (GHZ STATES). Here we show the connection between star graphs and single TCM plaquettes:

Using Lemma 2.2.2 and Remark 2.2.6 we showed (Example 3) that the ground state of any loop with N spins is LC-equivalent to $|\mathrm{GHZ}_N\rangle$ since it has the form

$$|\circ\rangle = \frac{1}{\sqrt{2}}\left[|+\rangle^{\otimes N} + |-\rangle^{\otimes N}\right]$$

Consider the hexagonal ($N = 6$) plaquette depicted in fig. 3.9. Constructing spanning trees is easy since deleting an arbitrary edge yields one. Consequently any graph state constructed this way must be a star graph as shown in the figure (this argument holds for N in general).
In [8] it was shown that star graphs and complete graphs[26] are LC-equivalent to GHZ states. Hence one can deduce that single plaquettes of any size must be LC-equivalent to GHZ states using the geometric rule and some knowledge about graph states.

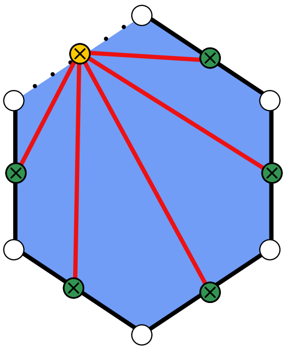


FIGURE 3.9 (Color online) : A hexagonal TCM plaquette. Deleting an arbitrary edge yields a spanning tree to construct a LC-equivalent graph state.

[26] Applying a local complementation to the center of the star yields the complete graph. Subsequent application of a local complementation to any vertex makes this vertex center of a star graph.

3.4 A side note: Spanning trees and local complementation

Theorem 3.2.2 states a geometric transformation rule in graph theoretic terms to find local Clifford equivalent graph states for a given Toric Code system. Since the graph obtained depends on the spanning tree chosen it follows that all graphs deduced from different spanning trees of the same Toric Code system must be LC-equivalent. In combination with the LC-rule developed in [2] this leads to the following (purely graph theoretic)

Lemma 3.4.1. *Given a (multi) graph $\mathcal{L}$ and two arbitrary spanning trees $\mathcal{L}_1'$ and $\mathcal{L}_2'$. Let $\mathcal{X}_1$ and $\mathcal{X}_2$ be the derived graphs as described in Theorem 3.2.2. Then $\mathcal{X}_1$ and $\mathcal{X}_2$ are equivalent under local complementations.*

This is a non-trivial statement from graph theory, derived indirectly using propositions developed for the description of graph states, i.e. for physical applications. As a proof of consistency it would be useful if this statement could be derived directly, without using the loop way over graph states. Hence the following proof:

Proof. We are going to prove Lemma 3.4.1 in two steps. First we show that two arbitrary spanning trees may be transformed into each other using a sequence of elementary transformations. In the second step it is shown that these elementary transformations lead to transformations of the derived graphs that can be achieved by a sequence of local complementations.

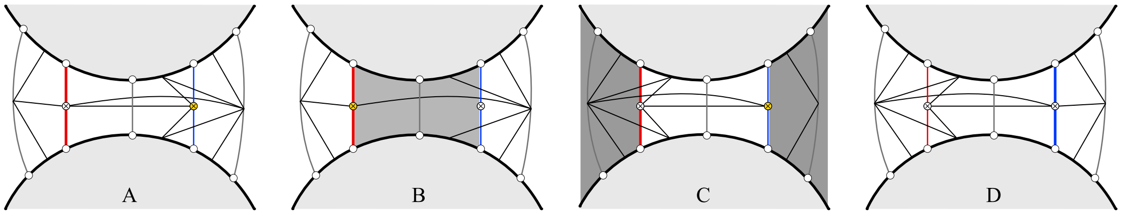


Figure 3.10 (Color online) : The sequence of local complementations as used in the proof of Lemma 3.4.1. The three steps A, B and C are described in the text.

1. Let $\mathcal{L}_1'$ and $\mathcal{L}_2'$ be arbitrary spanning trees of $\mathcal{L}$. We are going to transform $\mathcal{L}_1'$ into $\mathcal{L}_2'$ by a sequence of edge permutations $\sigma_{e,e'}$. If $\mathcal{L}_1' = \mathcal{L}_2'$ there is nothing to do. Thus let $\mathcal{L}_1' \neq \mathcal{L}_2'$. Then the difference of the edge sets is non-empty and one can choose $e \in E(\mathcal{L}_2') \setminus E(\mathcal{L}_1')$. Consider the graph $\mathcal{L}_1' + e$. This graph contains a cycle, more precisely: the fundamental cycle $\mathcal{C}_e$ with respect to $\mathcal{L}_1'$. Since $\mathcal{L}_2'$ is a tree it holds $\mathcal{C}_e \setminus E(\mathcal{L}_2') \neq \emptyset$. Choose $e' \in \mathcal{C}_e \setminus E(\mathcal{L}_2') \subseteq E(\mathcal{L}_1')$. The new graph $\sigma_{e,e'}\mathcal{L}_1' := \mathcal{L}_1' - e' + e$ is a tree since it does not contain $\mathcal{C}_e$. Furthermore it follows $|E(\mathcal{L}_2') \setminus E(\sigma_{e,e'}\mathcal{L}_1')| < |E(\mathcal{L}_2') \setminus E(\mathcal{L}_1')|$ since the new graph has one edge (e) more in common with $\mathcal{L}_2'$ than the old one. Successive application of this procedure aborts if $|E(\mathcal{L}_2') \setminus E(\sigma\mathcal{L}_1')| = 0$. Since both $\mathcal{L}_2'$ and $\sigma\mathcal{L}_1'$ are spanning trees it follows $\mathcal{L}_2' = \sigma\mathcal{L}_1'$ where σ denotes a finite combination of single edge permutations.

2. Let $\mathcal{L}_1'$ and $\mathcal{L}_2'$ be spanning trees such that $\mathcal{L}_2' = \sigma_{e,e'}\mathcal{L}_1'$. $\mathcal{X}_1$, $\mathcal{X}_2$ denote the derived graphs. If we identify the edges of $\mathcal{L}$ with the vertices of $\mathcal{X}_1$ and $\mathcal{X}_2$ it holds

$$\mathcal{X}_2 = \tau_e \tau_{e'} \tau_e \mathcal{X}_1$$

where $\tau_e, \tau_{e'}$ denote local complementations at vertices e and e'. This sequence is illustrated in fig. 3.10 where the vertices a local complementation is applied to are marked yellow. The spanning tree is denoted by bold lines (black and red in A) whereas elements of the fundamental cut $D_{e'}$ are drawn as thin lines (grey and blue in A). e and e' are denoted by blue and red lines, respectively. First note that by construction e has to be element of the fundamental cut $D_{e'}$ with respect to $\mathcal{L}_1'$. Then the sequence $\tau_e \tau_{e'} \tau_e$ acts as follows:

A τ_e is applied. Since e is connected to all vertices on the fundamental cycle $\mathcal{C}_e$ this operation yields a complete graph between these vertices[27]. This is denoted by a grey area in fig. 3.10 B.

B $\tau_{e'}$ is applied. Now several changes happen at once: First, the complete graph created in the previous step is broken up such that e' is now connected to all spins on the spanning tree that belong to the *future* fundamental cycle $\mathcal{C}_{e'}$. Secondly, all vertices $v \in D_{e'}$ with fundamental cycles $\mathcal{C}_v \setminus \mathcal{C}_e \neq \emptyset$ create a complete graph (denoted as dark grey area in fig. 3.10 C). Thirdly, all edges between these vertices and spins on the fundamental cycle $\mathcal{C}_e$ are switched. Therefore all vertices on the fundamental cut in the left grey area in C get *connected* to vertices on the tree in the white area whereas all vertices in the right grey area get *disconnected* from these vertices.

C τ_e is applied. The complete graph between all vertices on the fundamental cut marked dark grey in C is broken up such that e is connected to all spins on the *new* fundamental cut D_e. Hence we obtained $\mathcal{X}_2$.

Since the elementary spanning tree transformation $\sigma_{e,e'}$ can be translated into a sequence of local complementations acting on the derived graph and two spanning trees can be transformed into each other by elementary transformations $\sigma_{e,e'}$, the derived graphs of arbitrary spanning trees are LC-equivalent. This concludes the proof. ■

[27] Note that vertices on the spanning tree are not connected to each other directly.

Chapter 4

Toric Code systems and nonlocal graph states

4.1 Objective and overview

We saw that ground states of Toric Code Models are – as stabilizer states – equivalent to graph states under local Clifford operations. To this end let us consider the experimental point of view. It turns out that a sophisticated cooling mechanism as developed in [15] for cooling into TCM ground states is not required for *graph states.* If one allows local unitary operations on single qubits this yields an efficient procedure to obtain Toric Code ground states by cooling into graph states and subsequent LU-operations.

This is a somewhat abstract procedure since the spacial position of the spins is not considered. However, essential properties of Toric Code Models (e.g. the dependence of the surface topology and error correction procedures as well as the location and dynamics of elementary excitations) are closely connected to the *spacial structure* of the considered system. Therefore drawing a TCM as depicted in fig. 4.1 shows more than just abstract interactions if it comes to physical implementation; it shows how the spins should be arranged in order to deal with systems determined by *local interactions.*

Let us examine this point more precisely: As mentioned in the introduction the basis states of the protected space can be construed as the *ground states* of the Hamiltonian

$$H_{TCM} = - \sum_{s \in V(\mathcal{L})} A_s - \sum_{p \in P(\mathcal{L})} B_p \qquad (4.1)$$

where A_s are (generalised) star operators and B_p denote (generalised) plaquette operators defined by the faces of the graph $\mathcal{L}$ embedded on an orientable compact 2-manifold.

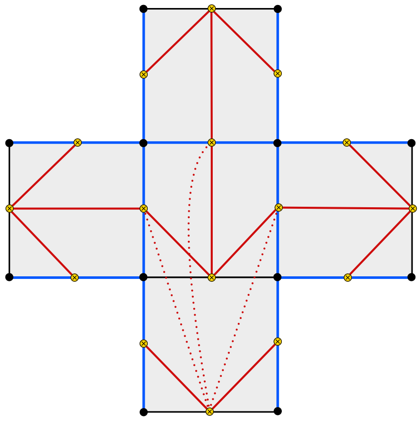


FIGURE 4.1 (Color online) : A Toric Code system with 16 spins (yellow) together with a LC-equivalent graph state based on the (blue) spanning tree. The dashed lines mark nonlocal graph state interactions with respect to the underlying TCM.

Hence physical implementations of such systems must deal with qubit interactions that are restricted to the support of single star and plaquette operators. If a Toric Code system is implemented by such interactions this defines an *adjacency relation* between the spins of the system. In this sense two spins e_1 and e_2 are *physically adjacent* iff there is either a star operator A_s such that $e_1, e_2 \in \text{supp}\,(A_s)$ or a plaquette operator B_p such that $e_1, e_2 \in \text{supp}\,(B_p)$.

In the majority of cases this adjacency relation implies spacial proximity if a physical implementation is considered but that is not the crucial point. The crux of the matter is if the *possibility of interaction* between two spins is given or not. The same argument holds for graph states where qubits are adjacent (as physically interacting two-level systems) if they are adjacent in the corresponding graph. This leads to the heart of the matter:

Let the *possibility of interaction* between the spins (i.e. the adjacency relation) be determined by the Toric Code system we tend to implement. Then the "simple" procedure explained above (constructing the ground state by LU-operations from an efficiently cooled graph state) is feasible only if there is an LU-equivalent graph state that can be implemented under the restriction of the adjacency relation. We will call such graph states *local* with respect to the Toric Code system under consideration. Thus Toric Code systems that are not LU-equivalent to at least one local graph state cannot be obtained by means of graph states in physical implementations[28].

To illustrate the introduced notion of physically adjacent spins consider the TCM depicted in fig. 4.1. One constructs a LC-equivalent graph state using the (blue) spanning tree and ends up with the red graph (continuous and dashed lines). Here the continuous edges represent *local* graph state interactions with respect to the underlying Toric Code system whereas the dashed edges are *nonlocal* since the adjacent vertices (spins) are neither connected by a common star nor a common plaquette operator. Hence *this* graph state is nonlocal. Since this is not the only LC-equivalent graph state the question arises whether there is *at least one* local representative in the corresponding LC-class (which would suffice). Therefore the main task of this chapter is the search for minimal Toric Code systems *without* local LC-equivalent graph states[29].

Finally, define the notion of *local graphs* formally as follows:

Definition 4.1.1: (Adjacency relation, Locality matrix). *Let G be a (simple) graph and $V(G)$ its vertex set.*

*(i) An **adjacency relation** $\sim_\Lambda$ on $V(G)$ is an arbitrary **irreflexive** and **symmetric** relation, i.e. there is no $v \in V(G)$ such that $v \sim_\Lambda v$ and for each pair $v, w \in V(G)$ it holds $v \sim_\Lambda w \Rightarrow w \sim_\Lambda v$.*

(ii) An adjacency relation is represented by an adjacency matrix $\mathbf{\Lambda}$ defined by

$$\Lambda_{v,w} := \begin{cases} 1\,, & \text{if } v \sim_\Lambda w \\ 0\,, & \text{otherwise} \end{cases} \tag{4.2}$$

*We call $\mathbf{\Lambda}$ the **locality matrix** defining an adjacency relation on $V(G)$. It is $\Lambda_{v,v} = 0$ for all $v \in V(G)$ and $\mathbf{\Lambda}^T = \mathbf{\Lambda}$ due to the irreflexive and symmetric property of $\sim_\Lambda$.*

*(iii) G is called **local** with respect to $\sim_\Lambda$ iff*

$$\mathbf{\Gamma}[G] * \overline{\mathbf{\Lambda}} = \mathbf{0} \tag{4.3}$$

where $*$ denotes the element-wise product of two matrices over $\mathbb{F}_2$ and $\overline{\mathbf{\Lambda}}$ is the complementation of $\mathbf{\Lambda}$ in $\mathbb{F}_2$, i.e. the matrix obtained by the substitution $1 \leftrightarrow 0$. In other words: G is local with respect to $\sim_\Lambda$ iff G is a subgraph of the graph defined by the adjacency matrix $\mathbf{\Lambda}$.

At first glance the element-wise product of matrices appears somewhat strange but since this test will be performed automatically by a `MATHEMATICA`-script it turns out to be a simple and fast technique to probe whether a given graph is local (in fact, this is a test for the subgraph property in general). To this end we may outline the algorithmic approach to test whether local graph states exist:

1. Use the given Toric Code system to extract the notion of *physically adjacent* spins and encode this information as locality matrix.

[28] If the coherent spacial exchange of particles is not feasible.

[29] In the following LC-equivalence instead of LU-equivalence is considered. See sec. 1.7 for further explanations.

2. Using an arbitrary spanning tree find a LC-equivalent graph state. Encode its structure as an adjacency matrix.

3. Generate the whole LC-class defined by this graph and check whether there are representatives that satisfy eq. (4.3).

4. If so, abort the algorithm since the Toric Code system in question proved local.

Consequently we have to investigate how to generate the whole equivalence class under local complementations of a simple graph.

4.2 Computing LC-orbits of graph states

In order to compute the whole equivalence class of a given graph state using `MATHEMATICA`, we have to compute the orbit of the group $\mathcal{T}$ generated by all local complementations τ_a acting on the edge space $\mathcal{E}(V)$ defined over a vertex set V (see def. 1.4.6). Thus our goal is

$$[G]_{\mathrm{LC}} = \mathcal{T}G \quad \text{where} \quad \mathcal{T} = \operatorname{span}\{\tau_a : \mathcal{E}(V) \to \mathcal{E}(V) \,|\, a \in V\} \tag{4.4}$$

and $G \in \mathcal{E}(V)$ denotes an arbitrary representative of the LC-class in question. As a check of consistency we are going to compute the size of some equivalence classes and if there are 2-colorable representatives. These results are compared to the data published by Hein *et al.* (Table II. in ref. [2]).

As described in [8] and originally derived in [25, 26] two graphs G and G' with adjacency matrices $\mathbf{\Gamma} \equiv \mathbf{\Gamma}[G]$ and $\mathbf{\Gamma}' \equiv \mathbf{\Gamma}[G']$ are equivalent under local complementations iff there exist diagonal matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D} \in \mathbb{F}_2^{N\times N}$ such that the non-linear condition

$$\mathbf{AD} + \mathbf{BC} = \mathbf{E}_N \tag{4.5}$$

and the system of linear equations

$$(\mathbf{\Gamma B} + \mathbf{D})\,\mathbf{\Gamma}' + (\mathbf{\Gamma A} + \mathbf{C}) = \mathbf{0} \tag{4.6}$$

hold over the Galois field $\mathbb{F}_2$.

It is easy to see that eq. (4.5) can be solved by the 6 combinations per dimension

$$(A_{ii}, D_{ii}, B_{ii}, C_{ii}) = \begin{cases} (1,1,0,0) \\ (1,1,0,1) \\ (1,1,1,0) \\ (0,0,1,1) \\ (0,1,1,1) \\ (1,0,1,1) \end{cases} \tag{4.7}$$

Consequently there are 6^N possible solutions for eq. (4.5). Since our final goal is the computation of the whole LC-class defined by a given graph G as representative, the following approach might be feasible for small N ($N \lesssim 12$):

1. Choose one of the 6^N solutions for eq. (4.5)

2. and solve the linear system (4.6) with respect to $\mathbf{\Gamma}'$ for a fixed adjacency matrix $\mathbf{\Gamma}$ describing the representative G.

Regarding step 1 we have to think about the structure of the solution space of eq. (4.6). The answer is given by the following

Lemma 4.2.1. *Equation (4.6) is soluble under the condition (4.5) if and only if* $\mathbf{\Gamma B} + \mathbf{D}$ *is invertible over* $\mathbb{F}_2$*. Therefore the solution set is empty or the solution is unique.*

Proof. We are going to show that the rank of the expanded $N \times 2N$-matrix $[\,\mathbf{\Gamma B} + \mathbf{D} \,|\, \mathbf{\Gamma A} + \mathbf{C}\,]$ is always N, i.e.

$$\operatorname{rank}[\,\mathbf{\Gamma B} + \mathbf{D} \,|\, \mathbf{\Gamma A} + \mathbf{C}\,] = N$$

Then it follows immediately that the equation

$$(\mathbf{\Gamma B} + \mathbf{D})\,\mathbf{\Gamma}' = (\mathbf{\Gamma A} + \mathbf{C})$$

is soluble only if $\operatorname{rank}[\,\mathbf{\Gamma B} + \mathbf{D}] = \operatorname{rank}[\,\mathbf{\Gamma B} + \mathbf{D} \,|\, \mathbf{\Gamma A} + \mathbf{C}\,] = N$. To prove our claim regarding the rank have a look at the expanded matrix in detail[30]

$$\left[\begin{array}{ccc|ccc} B_1\Gamma_{11} + D_1 & \cdots & B_N\Gamma_{1N} & A_1\Gamma_{11} + C_1 & \cdots & A_N\Gamma_{1N} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ B_1\Gamma_{N1} & \cdots & B_N\Gamma_{NN} + D_N & A_1\Gamma_{N1} & \cdots & A_N\Gamma_{NN} + C_N \end{array}\right] \tag{4.8}$$

where $A_i \equiv A_{ii}$ etc. for the sake of simplicity. To show that the rank of this matrix equals N in any case one can show that the standard basis of $\mathbb{F}_2^N$ can be constructed by appropriate sums of column vectors in (4.8). To construct the ith standard basis vector we must differentiate the following cases:

1. Let $A_i = D_i = 1$. If $B_i = 0$ the ith column of the left $N \times N$ block is already the basis vector we are looking for. If $B_i = 1$ we have $C_i = 0$. In this case add the ith column of the right block to the ith column of the left one. Since operations are performed modulo 2 this yields the required basis vector.

2. Let $B_i = C_i = 1$. If $A_i = 0$ the ith column of the right $N \times N$ block is already the basis vector we are looking for. If $A_i = 1$ we have $D_i = 0$. In this case add the ith column of the left block to the ith column of the right one. This yields the basis vector in question.

Using this algorithm one can construct the standard basis of $\mathbb{F}_2^N$ by linear combinations of column vectors. This concludes the proof. ∎

With Lemma 4.2.1 at hand our algorithmic approach reads as follows:

1. Choose one of the 6^N solutions for eq. (4.5).

2. Check whether $\det[\,\mathbf{\Gamma B} + \mathbf{D}] = 1$ or $\operatorname{rank}[\,\mathbf{\Gamma B} + \mathbf{D}] = N$, alternatively.

3. (a) If the matrix is non-singular, compute the solution

$$\mathbf{\Gamma}' = (\mathbf{\Gamma B} + \mathbf{D})^{-1}(\mathbf{\Gamma A} + \mathbf{C}) \tag{4.9}$$

   (b) If the matrix is singular skip all solutions of eq. (4.5) with the same matrices **B** and **D** (since the invertibility depends only on these matrices).

Step 3 (b) might speed-up the computation considerably, if non-singular matrices are rare (this is often the case). This is due to the fact that for each pair of matrices **B** and **D** there are 2^N pairs of matrices **A** and **C** that satisfy condition (4.5). Let p be the approximate rate of singular matrices, then the speed-up is estimated as

$$\delta \approx \frac{(1-p)\cdot 6^N + p\cdot 6^N 2^{-N}}{6^N} = (1-p) + \frac{p}{2^N} \overset{N\gg 1}{\approx} 1 - p \tag{4.10}$$

Nevertheless, the algorithm remains inefficient[31]. To my knowledge up to know there is no efficient (i.e. polynomial) algorithm to compute the whole LC-class of a given graph.

[30] Note that $\Gamma_{ii} = 0$ since $\mathbf{\Gamma}$ is the adjacency matrix of a simple graph. However, this property is not needed.

[31] Here "inefficient" means that the computation time scales exponentially with the number of edges involved. An "efficient" algorithm belongs to the class $\mathcal{O}(N^p)$.

4.3 Analysis of small Toric Code systems

Our goal, identifying minimal spin systems with nonlocal LC-equivalent graph states for regular lattices, is reached by applying the theory derived in the previous section to Toric Code systems that are complex enough. By "complex enough" we mean the following:

Consider a lattice with n-fold rotational symmetry in two dimensions and a single plaquette type[32] (i.e. $n = 3, 4, 6$). It is easy to see that at least $n + 1$ plaquettes are necessary to prevent at least one plaquette from having a peripheral boundary (i.e. an edge separating the interior from the non-tiled surrounding area). Thus at least 3 triangles are necessary to prevent the fourth, inner triangle from having a peripheral edge (see fig. 4.2 A). Analogously 4 squares are needed to surround the fifth one as shown in fig. 4.6 A. How does this observation apply to our problem of finding minimal spin systems for regular tessellations that cannot be described by local graphs?

To this end consider n connected tiles with n-fold symmetry that form a structure without a "hole". Consequently each tile has at least one peripheral edge. Delete one peripheral edge per tile. The edges remaining form a spanning tree of the system since there is no loop left. If a loop exists there would be at least one plaquette surrounded by it. Such a plaquette cannot exist since there must be at least one "cut" connecting the interior of each plaquette to the surrounding area.

In this case each fundamental cycle (i.e. the loops created by reinserting the deleted edges) coincides with the boundary of a certain plaquette. Using the graph theoretic rule derived in subsec. 3.2.2 one obtains a LC-equivalent graph state connecting only spins with a common plaquette. Therefore such systems are *always* local.

This is the reason why the systems depicted in figs. 4.2 A and 4.6 A are the smallest configurations for triangle and square tilings that may be nonlocal. Since the LC-class of a given graph always contains graphs with odd cycles which cannot be obtained by the rule mentioned above, the absence of local graphs obtained by this rule does not imply the absence of local graphs in the whole LC-class[33].

Results Using the `MATHEMATICA`-scripts described in appendix B based on the theory explained in sec. 4.2 the Toric Code systems depicted in figs. 4.2 A (`system 1`), 4.3 A (`system 2`), 4.4 A (`system 3`), 4.5 A (`system 4`) and 4.6 A (`system 5`) were analysed with respect to local graph states. The computations for `system 1`, `system 2` and `system 3` were performed with the script described in sec. B.1 since $N \leq 9$ is small enough. Hence we gained further information about the corresponding LC-classes (see Table 4.1). Since $N = 12$ for `system 4` we used the script described in sec. B.2 to probe for local graph states only (without gathering further information). The computation for `system 5` ($N = 16$) takes about 100 days on a modern personal computer. Since this is not feasible without using a supercomputer we cannot decide whether `system 5` is nonlocal. To face this problem a theoretical approach is taken in sec. 4.4.

No.	Solutions	Graphs	Unique graphs	Local graphs	2-col. graphs
1	3391488	6624	828 (66)	0 (0)	54
2	4218880	8240	8240 (763)	216 (31)	75
3	606208	2368	148 (6)	0 (0)	32
4	-	-	-	0 (0)	-
5	-	-	-	?	-

Table 4.1 (Color online) : Results of the computations using the `MATHEMATICA`-scripts described in appendix B. Bracketed values denote numbers of graphs up to graph isomorphisms.

Whereas `system 1` and `system 5` are interesting due to their properties described above, `system 3` and `system 4` are analysed for preparatory purposes (see subsec. 4.4.3). `system 2`

[32] Such tilings are called *platonic* or *regular tessellations*. There are only three of them: Triangular, square and hexagonal tiling.

[33] However, for the lack of counterexamples I conjecture that this implication holds. Since I was not able to prove it we cannot use it in the following.

was chosen to show that not every system featuring a plaquette surrounded by other plaquettes needs to be nonlocal. The results of the computations are listed in Table 4.1. **Solutions** lists the numbers of (not necessarily unique) solutions found for eq. (4.6) by testing all possible solutions for eq. (4.5). **Graphs** tells us how many of these solutions are adjacency matrices (thus describing simple graphs) and **Unique graphs** lists these numbers without duplicates. How many of these graphs are *local* is shown in the next column. Finally, **2-col. graphs** lists the numbers of 2-colorable graphs in the LC-class. The bracketed values denote the corresponding numbers of graphs up to graph isomorphisms.

The following facts are noteworthy:

- `system 1` is nonlocal. Hence it is the smallest nonlocal spin system for triangular tilings.
- `system 2` is local although there is a plaquette surrounded by other plaquettes and the graph obtained by the used spanning tree is nonlocal. However, if one chooses an appropriate spanning tree one finds immediately a local graph state representative[34].
- `system 3` is nonlocal. This will be used later on (see subsec. 4.4.3) to show that `system 5` is nonlocal as well.
- `system 4` is nonlocal. This will be approved in subsection 4.4.3 using the nonlocality of `system 1`.
- There are always 2-colorable representatives. This follows from the graph theoretic rule derived in section 3.2.2.

Hence we found the minimal nonlocal spin system for triangular tilings but the problem for square tilings remains unsolved. To overcome this obstacle the approach described in the next section will trace the nonlocality of `system 5` to the nonlocality of `system 3`.

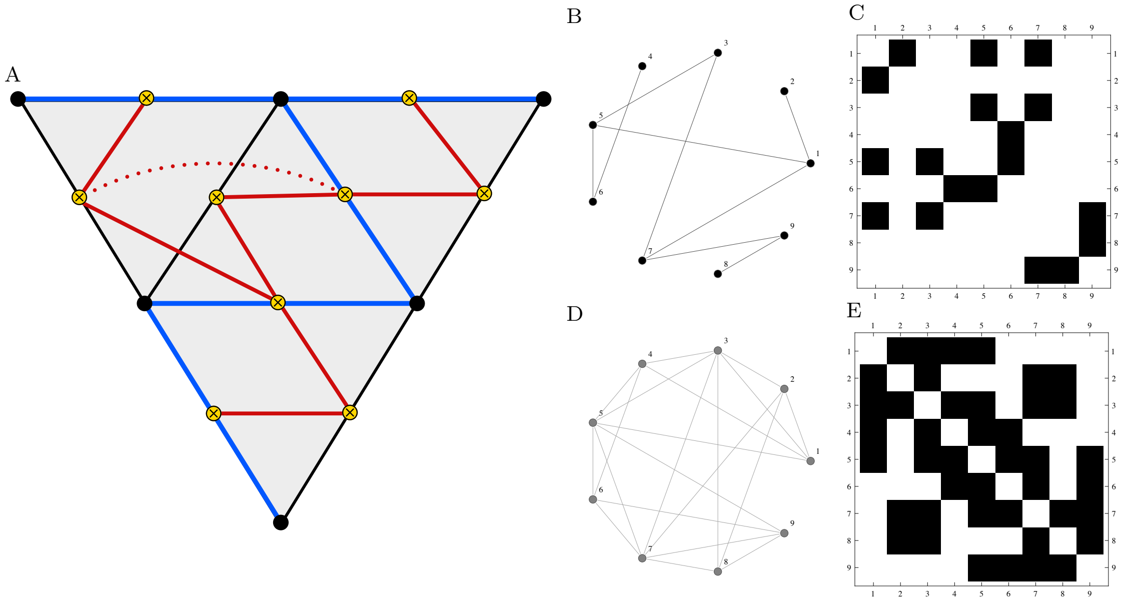


Figure 4.2 (Color online) : The LC-equivalent graph state representative of `system 1` (see A) is shown in B. The corresponding adjacency matrix is depicted in figure C and the locality matrix in E (black/white squares denote fields occupied by 1/0). The edge-maximal local graph is drawn in figure D. Obviously the graph state representative is nonlocal with respect to the adjacency relation induced by the underlying Toric Code system since B is not a subgraph of D. Computations show that this system is nonlocal.

[34] This is left as an exercise to the reader.

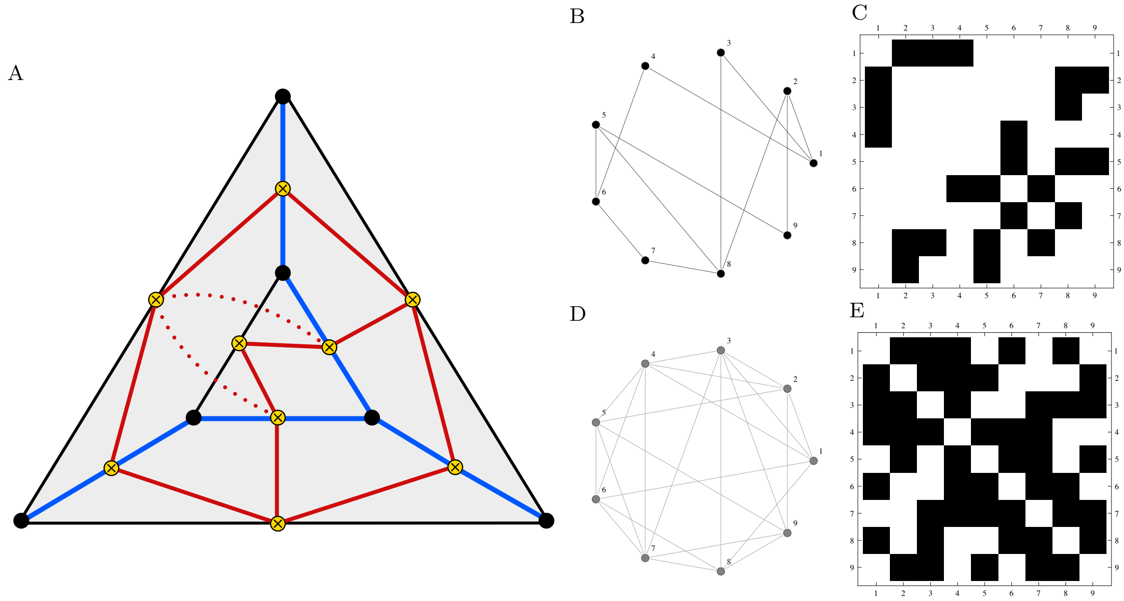


Figure 4.3 (Color online) : The LC-equivalent graph state representative of `system 2` (see A) is shown in B. The corresponding adjacency matrix is depicted in figure C and the locality matrix in E. The edge-maximal local graph is drawn in figure D. Obviously the graph state representative is nonlocal with respect to the adjacency relation induced by the underlying Toric Code system since B is not a subgraph of D. Nevertheless, this is a local Toric Code system as the computations show.

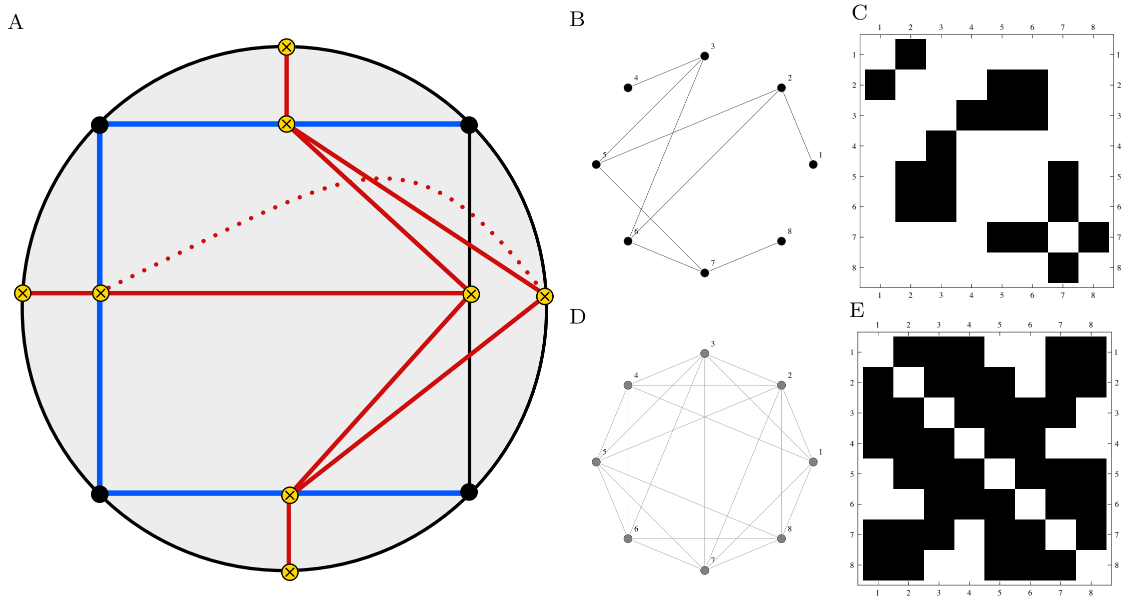


Figure 4.4 (Color online) : The LC-equivalent graph state representative of `system 3` (see A) is shown in B. The corresponding adjacency matrix is depicted in figure C and the locality matrix in E. The edge-maximal local graph is drawn in figure D. Obviously the graph state representative is nonlocal with respect to the adjacency relation induced by the underlying Toric Code system since B is not a subgraph of D. Computations show that this system is nonlocal.

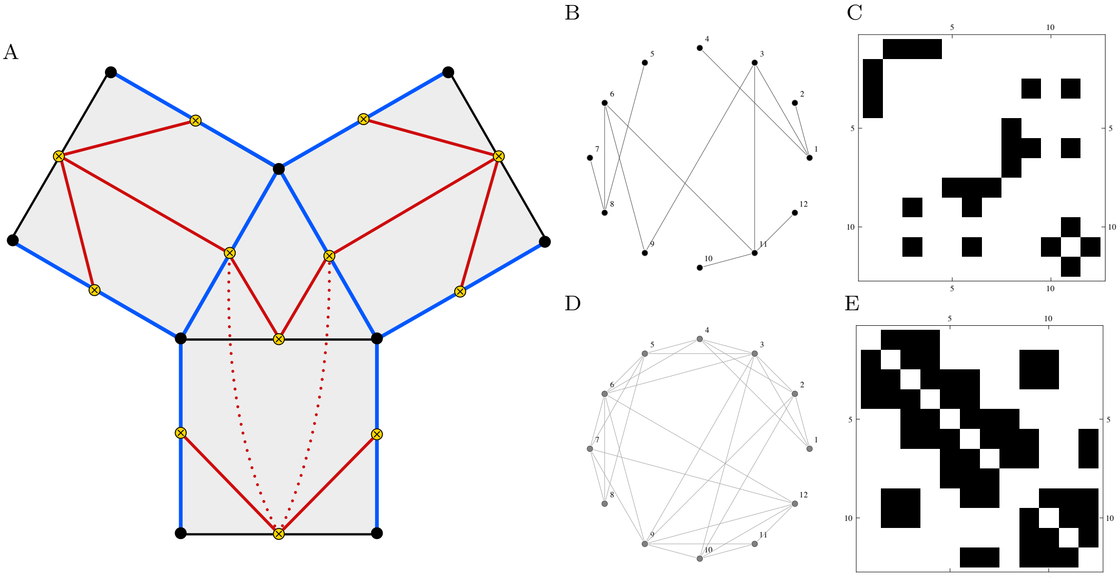


FIGURE 4.5 (Color online) : The LC-equivalent graph state representative of `system 4` (see A) is shown in B. The corresponding adjacency matrix is depicted in figure C and the locality matrix in E. The edge-maximal local graph is drawn in figure D. Obviously the graph state representative is nonlocal with respect to the adjacency relation induced by the underlying Toric Code system since B is not a subgraph of D. Computations show that this system is nonlocal.

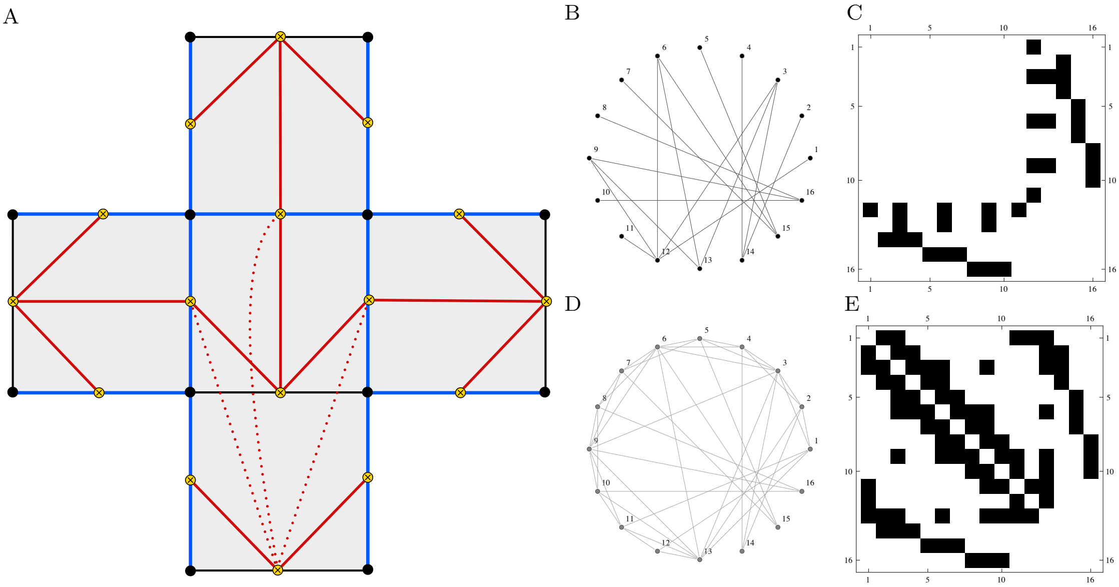


FIGURE 4.6 (Color online) : The LC-equivalent graph state representative of `system 5` (see A) is shown in B. The corresponding adjacency matrix is depicted in figure C and the locality matrix in E. The edge-maximal local graph is drawn in figure D. Obviously the graph state representative is nonlocal with respect to the adjacency relation induced by the underlying Toric Code system since B is not a subgraph of D. Whether this system is nonlocal cannot be decided by straightforward computation. In fact, it *is* nonlocal as shown in subsec. 4.4.3.

4.4 Reduction of complex Toric Code systems

As we saw in the previous section brute force computation of LC-classes is feasible only for small systems ($N \lesssim 12$) but becomes more and more time consuming for increasing numbers of spins. In order to decide whether `system 5` is nonlocal or not we have to find another, more “intelligent” procedure than brute force computation of the corresponding LC-class. As will become clear later on even the nonlocality of `system 4` can be derived this way.

As we will need some graph theoretic propositions regarding a special type of LC-classes, let us consider the mathematical point of view first.

4.4.1 On the structure of ε-symmetric LC-classes

Consider a simple graph which has at least one leaf (see subsec. 1.4.2 for basic definitions regarding graph theory). Let a and b be the outer and inner vertex, respectively. Then the statement of the following lemma is easy to verify.

Lemma 4.4.1. *Let $[G]_{\text{LC}}$ be the LC-class of a simple graph G such that there exist representatives with at least one leaf. If a and b denote the outer and inner vertex of a leaf, then the whole LC-class is invariant under the exchange of a and b. We denote this symmetry by*

$$\varepsilon_{ab} : \mathcal{E}(V) \to \mathcal{E}(V) \tag{4.11}$$

*and call $[G]_{\text{LC}}$ **ε-symmetric** with respect to a and b. Formally this reads $\varepsilon_{ab}[G]_{\text{LC}} = [G]_{\text{LC}}$.*

Proof. Let w.l.o.g. G_{ab} be a representative with a leaf and a be the outer vertex. Then it is easy to verify that the two local complementations

$$G_{ba} = \tau_a \tau_b G_{ab}$$

yield a graph G_{ba} such that b is the outer vertex and all edges that connected b in G_{ab} with vertices $V \setminus \{a, b\}$ now belong to a (see fig. 4.7). In other words: The graph G_{ba} is obtained from G_{ab} by exchanging the vertices a and b. Hence the symmetry $\varepsilon_{ab} G_{ab} = G_{ba}$ is realized by two local complementations. Now consider an arbitrary element $H_{ab} \in [G]_{\text{LC}}$. H_{ab} is reachable from G_{ab} by subsequent application of local complementations. If $\tau(a, b)$ denotes this chain of LC-operations then let $\tau(b, a)$ be the chain obtained by the substitution $\tau_a \leftrightarrow \tau_b$. One easily verifies that $H_{ba} = \tau(b, a)G_{ba} = \tau(b, a)\varepsilon_{ab}G_{ab} = \tau(b, a)\tau_a\tau_b G_{ab}$ is isomorphic to H_{ab} with the only difference that a and b are permuted. We write $H_{ba} = \varepsilon_{ab} H_{ab}$ and since $\varepsilon_{ab}^2 = \mathbb{1}$ it follows $\varepsilon_{ab}[G]_{\text{LC}} = [G]_{\text{LC}}$. ■

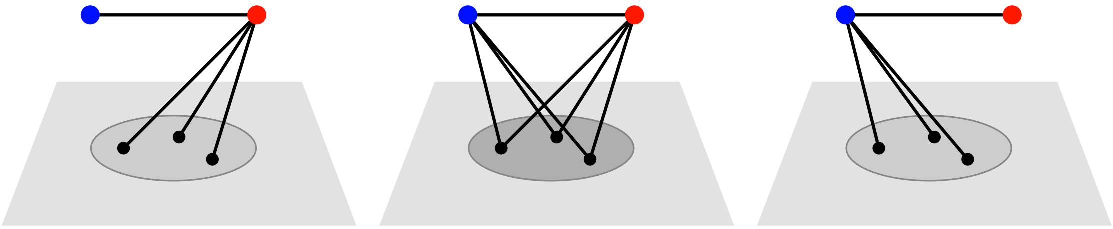


Figure 4.7 (Color online) : The exchange of two vertices is shown step by step as used in the proof in Lemma 4.4.1. The system on the left is transformed into the middle one by a local complementation at the red vertex. Subsequently the right graph is obtained by a local complementation at the blue vertex. Note that edges between the black vertices are restored during this sequence.

The symmetry originally shown for the special leaf-graphs G_{ab} and G_{ba} “spreads” through the whole equivalence class and leads to some useful properties regarding LC-classes represented by such graphs. Before analysing this structure let us discuss the following statement in detail:

Lemma 4.4.2. *Let $G_{ab} \in \mathcal{E}(V)$ be a simple graph with at least one leaf. a denotes the outer and b the inner vertex. Consider the subgroup*

$$\mathcal{T}_a := \operatorname{span}\{\tau_i \,:\, \mathcal{E}(V) \to \mathcal{E}(V) \,|\, i \in V \setminus \{a\}\} \tag{4.12}$$

of all local complementations on vertices $V \setminus \{a\}$. Then it holds

$$\forall\, \tau \in \mathcal{T}_a \,:\, \tau G_{ab} - a = \tau\,(G_{ab} - a) \tag{4.13}$$

In words: Deleting the leaf (a and its adjacent edge) from G_{ab} and transforming this graph by a sequence of local complementations τ acting on vertices $V \setminus \{a\}$ yields the same graph as if these operations were performed on G_{ab} (without deleting the leaf) and deleting a with all its adjacent edges (which may be more than just one) afterwards.

Proof. It is easy to see that all changes regarding edges between vertices in $V \setminus \{a\}$ occur in the same way if or if not a connection to a is present. Therefore the only differences between τG_{ab} and $\tau\,(G_{ab} - a)$ affect edges from vertices in $V \setminus \{a\}$ to a itself. Since these edges are deleted $(\tau G_{ab} - a)$ there is no difference left and it holds $\tau G_{ab} - a = \tau\,(G_{ab} - a)$. ■

Lemma 4.4.2 allows statements about the LC-class $[G_{ab}-a]_{\mathrm{LC}}$. More precisely: All these graphs in $[G_{ab}]_{\mathrm{LC}}$ yield graphs in $[G_{ab} - a]_{\mathrm{LC}}$ by deleting a and its adjacent edges that are reachable from G_{ab} by concatenations of local complementations in $\mathcal{T}_a$. If one can show that all graphs in $[G_{ab}]_{\mathrm{LC}}$ may be linked to each other by operations in such subgroups, this yields a powerful procedure to compute LC-classes of special subgraphs.

This idea leads to the following proposition for simple graphs with leafs.

Proposition 4.4.3: (ε-symmetric LC-classes). *Let G_{ab} be a simple graph with at least one leaf where a and b denote the outer and inner vertex, respectively. Set $[G]_{\mathrm{LC}} \equiv [G_{ba}]_{\mathrm{LC}} = [G_{ab}]_{\mathrm{LC}}$. Then it holds*

$$\forall\, H \in [G]_{\mathrm{LC}}\, \exists\, (H'_a \in [G_{ab} - a]_{\mathrm{LC}} \,\vee\, H'_b \in [G_{ba} - b]_{\mathrm{LC}}) \,:\, (H - a = H'_a \,\vee\, H - b = H'_b) \tag{4.14}$$

In words: If a LC-class is ε-symmetric with respect to a and b then deletion of a or b (in some cases it does not matter which one is deleted) yields a graph in $[G_{ab} - a]_{\mathrm{LC}}$ or $[G_{ba} - b]_{\mathrm{LC}}$. In a nutshell: Every graph in $[G]_{\mathrm{LC}}$ has at least one subgraph in $[G_{ab} - a]_{\mathrm{LC}}$ or $[G_{ba} - b]_{\mathrm{LC}}$.

The reader may notice that Proposition 4.4.3 could be useful for the reduction of large Toric Code systems to smaller ones while keeping track of the corresponding equivalence classes of graph states. In fact, this will be explained in detail in the subsequent section. Prior to that we have to prove Proposition 4.4.3.

Proof. Let G_{ab} be a representative with outer and inner vertex a and b. Then one can partition the LC-class $[G_{ab}]_{\mathrm{LC}}$ in the following way:

- All graphs with a leaf belong to A iff a is the outer and b the inner vertex.
  Therefore $G_{ab} \in A$.
- All graphs with a leaf belong to B iff b is the outer and a the inner vertex.
  Therefore $G_{ba} \in B$.
- All graphs H such that the following holds belong to C

  $$\{a,b\} \in E(H) \quad \text{and} \quad \forall\, v \in V(H) \setminus \{a,b\} \,:\, \{a,v\} \in E(H) \Leftrightarrow \{b,v\} \in E(H)$$

  Due to the second statement these graphs are totally symmetric with respect to a and b, $\varepsilon_{ab} H = H$ that is.

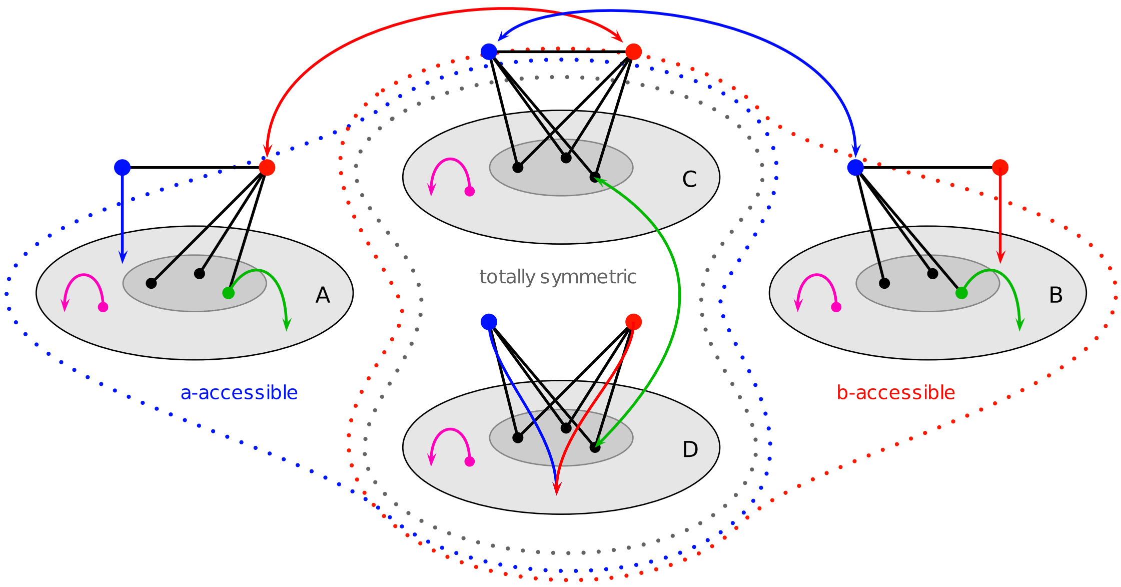


Figure 4.8 (Color online) : Diagrammatic overview showing the structure of ε-symmetric LC-classes as used in the proof of Proposition 4.4.3. Arrows denote different classes of transformations according to their color: γ-operations (magenta), ξ-operations (green), a-operations (blue) and b-operations (red).

- All graphs H such that the following holds belong to D

$$\{a,b\} \notin E(H) \quad \text{and} \quad \forall\, v \in V(H) \setminus \{a,b\} \,:\, \{a,v\} \in E(H) \Leftrightarrow \{b,v\} \in E(H)$$

  These graphs are totally symmetric and the only difference to type C graphs is the missing edge between a and b.

These four types of graphs are depicted in fig. 4.8. Note that the number of edges between a or b and the vertices $V(G) \setminus \{a,b\}$ (denoted by light grey discs) is not constant within one class in general. That the described classification of graphs is a *partition*, i.e. $[G_{ab}]_{\text{LC}} = A \dot\cup B \dot\cup C \dot\cup D$ has to be shown. To this end we introduce a classification of local complementations $\tau_v \in \mathcal{T}$ which may depend on the graph it is operating on. We call τ_a and τ_b an *a- and b-operation*. Local complementations τ_v where $v \in V(G) \setminus \{a,b\}$ and $\{a,v\},\{b,v\} \notin E(G)$ are called *γ-operations* and LC-operations τ_v where $v \in V(G) \setminus \{a,b\}$ and $\{a,v\} \in E(G) \vee \{b,v\} \in E(G)$ are called *ξ-operations*. Obviously every local complementation τ_v , $v \in V(G)$ belongs (depending on the graph it is applied to) exactly to one of these four classes. Let us examine how these four classes of local complementations and the four classes of graphs introduced above collude:

- First note that γ-operations cannot alter the class a graph belongs to since all edges used for the classification of graphs are unaffected under γ-operations. In fig. 4.8 this is denoted by magenta arrows pointing from each set into itself.
- Clearly, a-operations act trivially on A. Applied to an element of C it deletes all edges between b and $V(G) \setminus \{a,b\}$ and yields a graph that belongs to B. Since $\tau_a^2 = \mathbb{1}$ this is true for the opposite direction as well. Finally, graphs that belong to D cannot leave this set since no edge between a and b is created. These connections are denoted by blue arrows in fig. 4.8.
- Due to the symmetry with respect to a and b all statements for τ_a hold for τ_b (i.e. b-operations) as well. These relations are denoted by red arrows. Note that τ_a equals τ_b if they are applied to elements in D.

- ξ-operations act on vertices denoted by dark grey circles in fig. 4.8 (vertices that are connected to a or b or both of them). Obviously such operations cannot manipulate leafs. Therefore ξ-operations cannot leave the sets A and B (but they may alter the number of edges connecting a or b to vertices in $V(G) \setminus \{a, b\}$). Acting on elements in C a ξ-operation deletes the edge between a and b, hence converting type C graphs into type D graphs and vice versa. These relations are denoted by green arrows.

Consider the following situation: Starting from $G_{ab} \in A$ one tries to generate as much elements in $[G_{ab}]_{\text{LC}}$ as possible by using local complementations in $\mathcal{T}_a$. Therefore one is allowed to use b- γ- and ξ-operations to generate graphs. In terms of fig. 4.8 the "path" cannot use the blue arrows. Note that the only a-operation that cannot be omitted connects C and B (in A it acts trivially and in D it may be replaced by a b-operation). We call graphs in this orbit $\mathcal{T}_a G_{ab}$ *a-accessible*. In conclusion it holds $\mathcal{T}_a G_{ab} \subseteq [G]_{\text{LC}} \setminus B$. Analogue arguments yield $\mathcal{T}_b G_{ba} \subseteq [G]_{\text{LC}} \setminus A$ for *b-accessible* graphs starting from $G_{ba} \in B$.

We are now going to show that the more stricter relations $\mathcal{T}_b G_{ba} = [G]_{\text{LC}} \setminus A$ and $\mathcal{T}_a G_{ab} = [G]_{\text{LC}} \setminus B$ hold. To this end consider an arbitrary "path" jumping from class to class w.l.o.g. starting at G_{ab} and ending at $H \in [G]_{\text{LC}} \setminus B$, formally described by

$$H = \prod_i \tau_i G_{ab} \quad \text{where} \quad \tau_i \in \mathcal{T}$$

and depicted in fig. 4.9 A. In general there are some a-operations in this chain. Since neither G_{ab} nor H are in B there is an even number of them (in fig. 4.9 A there are two, denoted by blue jumps). We are restricted to non-a-operations in $\mathcal{T}_a$, though. We may adhere to this restriction by "swapping" the a-jumps and the corresponding γ- and ξ-operations in B to b-jumps and the same γ- and ξ-operations in A. This is shown in fig. 4.9 B.

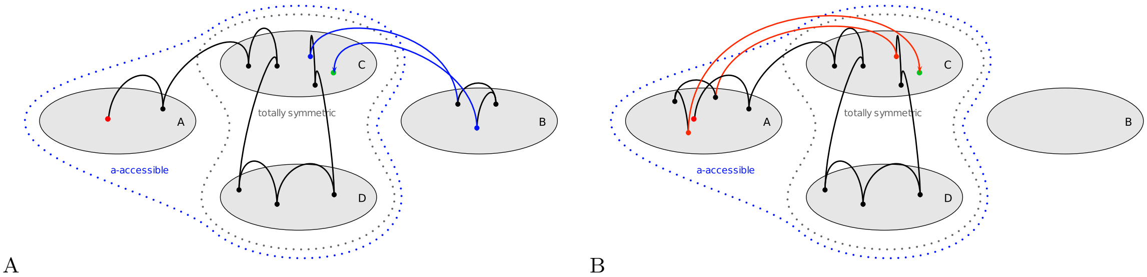


FIGURE 4.9 (Color online) : An arbitrary path from G_{ab} (red dot) to H (green dot) is depicted in A. In B the "forbidden" (blue) path using a-operations is swapped to A (red jumps) using b-operations instead. Now the whole path from G_{ab} to H is a-accessible.

It is easy to see that jumping to A by b-operations instead and subsequently applying the same γ- and ξ-operations yields a graph in A that resembles the graph obtained in B before jumping back to C with the only difference that a and b are permuted. However, since the second application of τ_b yields a totally symmetric graph in C this graph has to be the same one as formerly obtained by the second τ_a operation.

Consequently we found a procedure that allows us to reach any $H \in [G]_{\text{LC}} \setminus B$ by local complementations in $\mathcal{T}_a$ from G_{ab} and due to symmetry each $H' \in [G]_{\text{LC}} \setminus A$ may be reached by exclusive application of $\mathcal{T}_b$ operations from G_{ba}[35]. Thus we showed $\mathcal{T}_b G_{ba} = [G]_{\text{LC}} \setminus A$ and $\mathcal{T}_a G_{ab} = [G]_{\text{LC}} \setminus B$.

To conclude the proof recall Lemma 4.4.2. Consider $H \in [G]_{\text{LC}}$. Assume without loss of generality that $H \notin B$ (the following argument holds for $H \notin A$ analogously). Then we showed that $H \in \mathcal{T}_a G_{ab}$ where G_{ab} is a leaf-graph with outer vertex a. Choose $\tau \in \mathcal{T}_a$ such that $H = \tau G_{ab}$.

[35] It is noteworthy that elements in A may be reached *only* from G_{ab} and elements in B *only* from G_{ba} whereas this procedure shows that elements in C or D may be reached from both G_{ab} and G_{ba} using $\mathcal{T}_a$ or $\mathcal{T}_b$, respectively.

According to Lemma 4.4.2 it follows that $H - a = \tau G_{ab} - a = \tau\,(G_{ab} - a)$. Set $H'_a := \tau\,(G_{ab} - a) \in [G_{ab} - a]_{\mathrm{LC}}$ and we have $H - a = H'_a \in [G_{ab} - a]_{\mathrm{LC}}$. Which was to be proven.

■

Remark 4.4.4. *Proposition 4.4.3 shows that any graph $H \in [G_{ab}]_{\mathrm{LC}}$ (where G_{ab} is a leaf-graph with outer vertex a) has either a subgraph in $[G_{ab} - a]_{\mathrm{LC}}$ or in $[G_{ba} - b]_{\mathrm{LC}}$.*

To illustrate the structure of an ε-symmetric LC-class and the subgraph property mentioned above an example with 5 vertices is given in appendix C.

4.4.2 Application to the locality problem

In order to apply the knowledge we gained in the previous section to our physical problem (the locality of graph states) we have to study the effect of *edge transformations* on the adjacency relation $\sim_\Lambda$. To this end a further definition is required:

Definition 4.4.5: (STRICTNESS). *Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be (multi) graphs embedded into a surface, thus defining Toric Code Models with plaquettes $P\,(\mathcal{L}_1)$ and $P\,(\mathcal{L}_2)$. Furthermore let $E\,(\mathcal{L}_2) \subseteq E\,(\mathcal{L}_1)$. Thus $\mathcal{L}_2$ is obtained from $\mathcal{L}_1$ by removing some edges (spins) and transforming the resulting graph arbitrarily. Let Λ_1 and Λ_2 denote the induced adjacency relations. Then Λ_1 is called* ***stricter*** *than Λ_2 ($\Lambda_1 \preccurlyeq \Lambda_2$) iff*

$$\forall\, e_1, e_2 \in E\,(\mathcal{L}_2) \;:\; e_1 \nsim_{\Lambda_2} e_2 \Rightarrow e_1 \nsim_{\Lambda_1} e_2 \tag{4.15}$$

For an illustration of this notion the following example may be helpful:

Example 9: (STRICTNESS). Consider the Toric Code system depicted in fig. 4.10 A. The plaquette and star operators define an adjacency relation. E.g. the spins 10 and 12 are physically adjacent due to a common plaquette whereas 10 and 3 are adjacent due to a common star operator. Furthermore the spins 10 and 4 are *not* adjacent since there are neither common plaquette nor star operators.

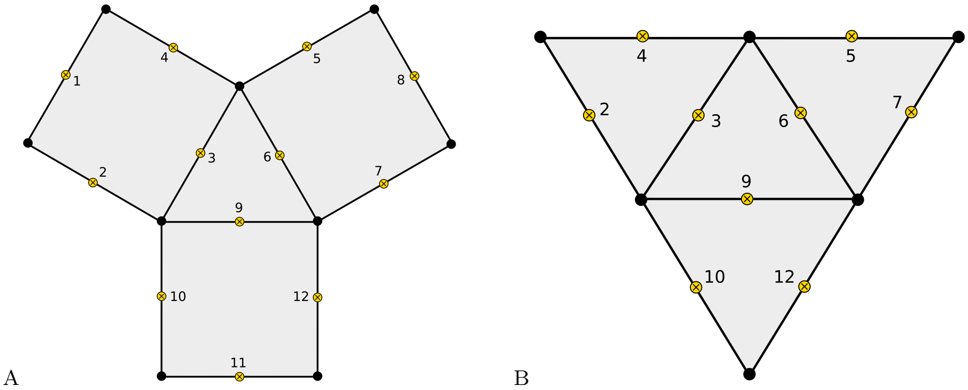


FIGURE 4.10 (Color online) : The Toric Code system in A is transformed into the system depicted in B by contracting the edges 1, 8 and 11, thus deleting three spins. One verifies easily that all spins that are *not* physically adjacent in the reduced system (B) are *not* physically adjacent in the original system (A).

Contracting the edges 1, 8 and 11 yields the system depicted in fig. 4.10 B which has only 9 spins left. It is now easy to verify that all spins that are *not* physically adjacent in the reduced system are *not* physically adjacent in the larger system. Therefore the adjacency relation of the system in A is stricter than the relation induced by the reduced system in B.

After these physical (Definition 4.4.5) and mathematical (Proposition 4.4.3) preliminaries we are now able to state and prove the central theorem of this section:

Theorem 4.4.6: (TCM reduction). *Let $\mathcal{L}$, $\mathcal{L}_a$ and $\mathcal{L}_b$ be (multi) graphs embedded into a surface, thus defining three Toric Code systems. Furthermore let Λ, Λ_a and Λ_b denote the induced adjacency relations. In addition*

(i) System $\mathcal{L}$ is stricter than the other systems, i.e. $\Lambda \preccurlyeq \Lambda_a$ and $\Lambda \preccurlyeq \Lambda_b$.

(ii) There is a leaf-graph G_{ab}, describing a LC-equivalent graph state of system $\mathcal{L}$, such that $G_{ab} - a$ and $G_{ba} - b$ describe LC-equivalent graph states of $\mathcal{L}_a$ and $\mathcal{L}_b$, respectively.

(iii) The Toric Code systems $\mathcal{L}_a$ and $\mathcal{L}_b$ are nonlocal.

Then the Toric Code System defined by $\mathcal{L}$ is nonlocal.

Proof. In order to show that the Toric Code system defined by $\mathcal{L}$ is nonlocal one has to find at least one nonlocal edge for each element in $[G_{ab}]_{\mathrm{LC}}$ with respect to the adjacency relation Λ. Let $H \in [G_{ab}]_{\mathrm{LC}}$ be an arbitrary graph representing a LC-equivalent graph state. According to Proposition 4.4.3 each element in $[G_{ab}]_{\mathrm{LC}}$ has at least one subgraph in $[G_{ab} - a]_{\mathrm{LC}}$ or $[G_{ba} - b]_{\mathrm{LC}}$. W.l.o.g. let $H'_a \in [G_{ab} - a]_{\mathrm{LC}}$ be such a subgraph, i.e. $H'_a \subseteq H$. Since H'_a describes a LC-equivalent graph state of system $\mathcal{L}_a$ (see (ii)) there is at least one edge $e = \{v_1, v_2\} \in E(H'_a) \subseteq E(H)$ such that the vertices[36] are physically not adjacent, i.e. $v_1 \not\sim_{\Lambda_a} v_2$ (see (iii)). Since system $\mathcal{L}$ is stricter than $\mathcal{L}_a$ (see (i)) it follows immediately $v_1 \not\sim_{\Lambda} v_2$. Thus we found an edge e in $E(H)$ that represents a nonlocal interaction in the larger system $\mathcal{L}$. Since H was chosen arbitrarily this shows the nonlocality of $\mathcal{L}$. ■

Applications of Theorem 4.4.6 are given in the next subsection.

4.4.3 Examples

In the following we are going to show that the nonlocality of `system 5` can be traced to the nonlocality of `system 3` which has been shown in sec. 4.3 by brute force computations. Furthermore the nonlocality of `system 4` is reviewed using Theorem 4.4.6 and the nonlocality of `system 1`. Hence the brute force analysis of `system 4` was not necessary.

Example 10: (System 4). In fig. 4.11 A the Toric Code `system 4` is shown – this is system $\mathcal{L}$. The leaf-graph G_{ab} is drawn with red lines and denotes a LC-equivalent graph state since it is constructed using the (blue) spanning tree. One obtains the leaf-graph G_{ba} by replacing the red edges in the upper left square with the green ones.

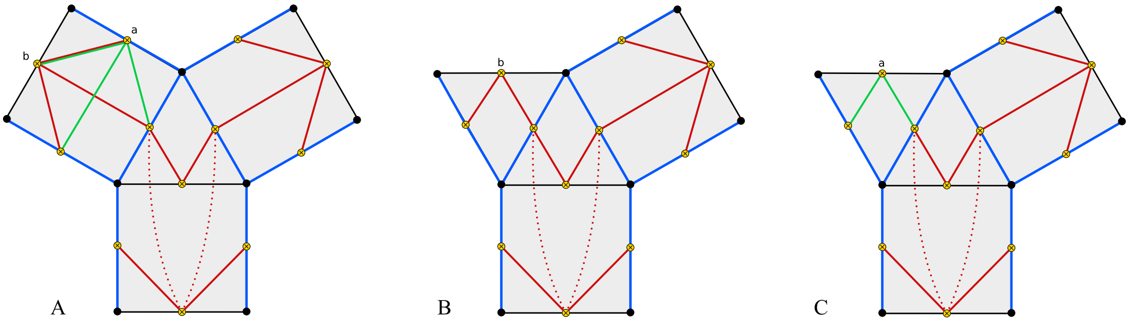


Figure 4.11 (Color online) : `system 4` is depicted in figure A. The leaf-graph G_{ab} is drawn with red lines whereas the leaf-graph G_{ba} is obtained by replacing the red lines in the upper left square with the green ones. Deleting a (b) and contracting the corresponding edge of the Toric Code system yields the system depicted in B (C). Here both systems $\mathcal{L}_a$ (fig. B) and $\mathcal{L}_b$ (fig. C) are equivalent up to a permutation of a and b.

Deleting a and b yields the graphs $G_{ab} - a$ and $G_{ba} - b$, respectively. These are drawn in fig. 4.11 B and C. Obviously they are equivalent up to a permutation of a and b. Furthermore

[36] Spins are *vertices* with respect to graph states and *edges* with respect to Toric Code systems.

they represent graph states that are LC-equivalent to the Toric Code systems drawn beneath the graphs $G_{ab} - a$ and $G_{ba} - b$. Therefore the systems $\mathcal{L}_a$ and $\mathcal{L}_b$ coincide. It is easy to check that the adjacency relation of the Toric Code system in A is stricter than the relations of the systems in B and C since we derived them by an edge contraction (see Example 9).

Assume that the Toric Code system depicted in B and C is nonlocal. According to Proposition 4.4.6 we find that the original system in A is nonlocal as well. It is now straightforward to show that this procedure works also for the remaining two squares. In the end one obtains a sequence of Toric Code systems (the original one, *one* triangle instead of a square, *two* triangles instead of two squares and `system 1`) related by a chain of implications. Since the first system in this chain (`system 1`) proved nonlocal, this implies at once the nonlocality of all other members in the chain. Especially we showed the nonlocality of `system 4` approving the result obtained previously by brute force computations.

Example 11: (System 5). As mentioned previously it is not feasible to show the nonlocality of `system 5` by computation of the whole LC-class for a corresponding graph state. On a modern personal computer this computation takes roughly 100 days since the system in question has 16 spins and the algorithm described in sec. 4.2 scales exponentially with the number of spins.

Therefore – in contrast to the previous example – the reduction to smaller spin systems (in this case we reduce `system 5` to `system 3`) is the *only* possibility to prove the nonlocality of the considered system.

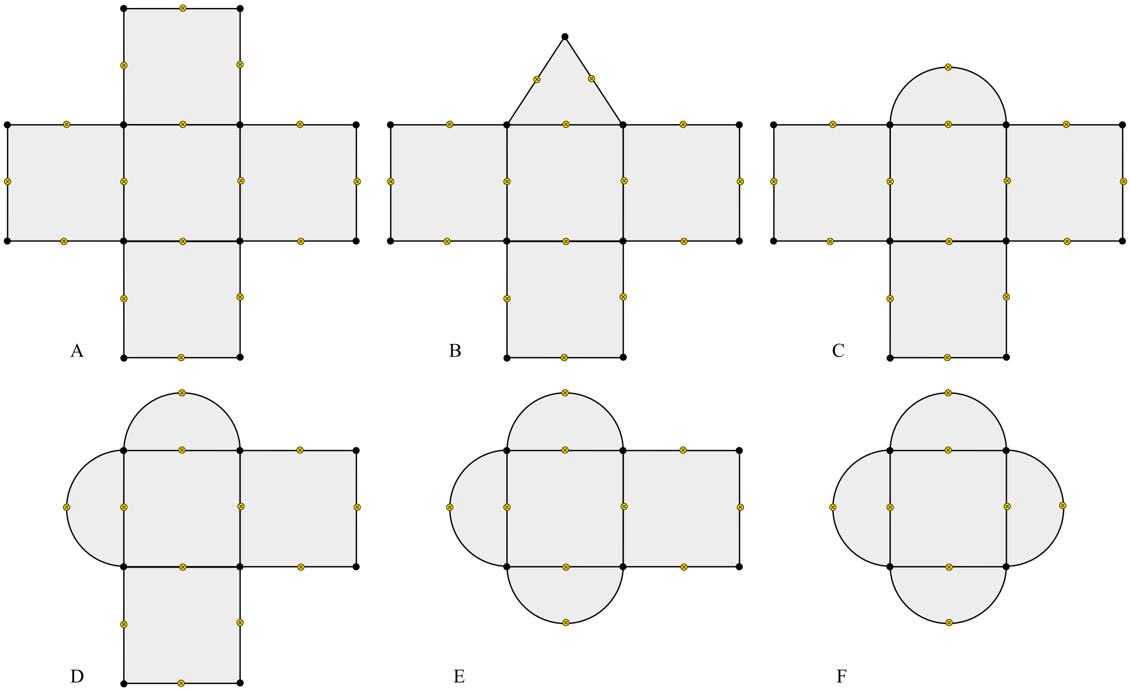


Figure 4.12 (Color online) : The original `system 5` is depicted in A. By reducing the upper square twice (as described in the previous example) one obtains successively the systems depicted in B and C. This procedure is repeatedly applied to the remaining squares (D,E) and leads to the Toric Code system depicted in F. Since this is `system 3` the nonlocality of the whole series of systems is proved.

`system 5` is depicted in fig. 4.12 A. The elementary steps to reduce the squares attached to the central square resemble the procedure described in the previous example. After reducing the upper square to a triangle by deleting one spin[37] the same procedure may be used to get rid of

[37] In all these cases the systems $\mathcal{L}_a$ and $\mathcal{L}_b$ coincide since spin a and b both belong to a "path" that was divided by several star operators. Therefore contracting the corresponding edge yields in both cases the *same* system which is furthermore LC-equivalent to the graph state described by $G_{ab} - a$ or $G_{ba} - b$, respectively. Thus applying Theorem 4.4.6 is comparatively simple in the considered cases.

another spin. This yields the system depicted in C[38]. Subsequently one applies this procedure to the remaining three squares (D,E). Finally one obtains the system shown in F. Since this is `system 3` (which was shown to be nonlocal by computations in sec. 4.3) we showed that all other systems constructed in the chain are nonlocal as well. Especially the last element of the chain – `system 5` – is proved to be nonlocal.

[38] To prevent misconceptions: Deleting the *last* spin would not make any sense since the prerequisites of Theorem 4.4.6 were violated (the resulting system is local). For we cannot deduce anything from a reduced *local* system, one realizes that the described approach is similar to estimating expressions in mathematics where one often "overshoots the mark".

Conclusion

In the first part of this thesis it was shown how Toric Code ground states may be derived from the ground states of their subsystems. We saw that the one-vertex connection of two subsystems leads to product states without entanglement between the two systems whereas connecting them twice results in entangled states due to the newly created loop.

In the second part it was shown how local Clifford equivalent graph states can be obtained from given Toric Code stabilizers in the binary framework. To this end we employed an algorithm known from the theory of stabilizer states. Subsequently it was shown that this transformation may be achieved without any calculation using spanning trees of the considered Toric Code system. A connection to graph theory was established since the derived rule can be expressed in terms of *fundamental cycles* and *fundamental cuts*.

In the last section an algorithmic approach was taken to determine whether for a given Toric Code system LC-equivalent graph states exist that feature only spin-spin interactions that are also present in the underlying Toric Code. We computed the smallest setting built from triangular plaquettes that cannot be prepared from local graph states by means of local Clifford operations. In addition it was shown how the nonlocality of certain larger Toric Code systems may be derived from the nonlocality of smaller ones. Using this approach we found the minimal spin setting on a generic square lattice that yields Toric Code ground states without LC-equivalent local graph states. Furthermore it is strongly believed that the conclusions stated above hold for the larger class of local unitary operations as well.

Appendix A

List of small Toric Code systems

The following four tables (A.1, A.2, A.3 and A.4) list several small Toric Code systems with up to 6 spins and some of their properties. These systems are symbolized by their stabilizer where continuous black lines denote spins, small circles mark star operators and blue areas indicate where plaquette operators exist. The degeneracy γ is computed using the relation given by Proposition 1.2.2. To observe anyonic statistics it is necessary to excite both m and e particles and move one of them around the other. Whether such operations are (rudimentary) possible is noted in column **AE**. If moving an e particle around a m excitation is possible but this path consists of less than three spins this is indicated by $(\cdot)$.

If the degeneracy is non-trivial (i.e. $\gamma > 1$) one obtains the topological quantum numbers v_{top} measuring the z-type loop operators. Whether the support of such operators includes spins that cannot interact via elementary star and plaquette operators, is indicated by the column **NL**. Such operators are called *nonlocal*[39]. The ground states were computed using the calculus developed in section 2.2. The symbols used for the sake of brevity are defined in sections 2.2 and 1.1. In column **LU/S** LC-equivalent graph states are depicted. Since $N \leq 7$ the equivalence classes with respect to local unitaries (LU) and local Clifford operations (LC) coincide. Note that all states that belong to different configurations of plaquette operators are LC-equivalent and therefore described by the same graph state. By omitting spin labels qubit SWAP operations are implied.

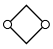
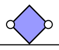
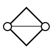
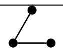
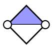
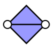
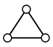
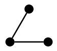
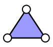
No.	N	**Setting**	γ	**AE**	**NL**	**GS** $\mathcal{N}^{-1}\|\text{GS}_i\rangle$, $1 \leq i \leq \gamma$	**LU/S**	**Comment**
1	1	○—○	2^0	no	no	$\|+\rangle$	•	single qubit
2	2	○—○—○	2^0	no	no	$\|+\rangle\|+\rangle$	• •	biseparable
3	2		2^1	no	no	$\|\Phi^+\rangle$,$\|\Psi^+\rangle$	•—•	Bell states
4	2		2^0	(yes)	no	$\|\Phi^+\rangle$	•—•	Bell state
5	3		2^2	no	no	$\|\Psi^+\rangle\|+\rangle \pm \|\Psi^-\rangle\|-\rangle$, $\|\Phi^+\rangle\|+\rangle \pm \|\Phi^-\rangle\|-\rangle$		
6	3		2^1	(yes)	no	$\|\Phi^+\rangle\|+\rangle \pm \|\Phi^-\rangle\|-\rangle$		
7	3		2^0	(yes)	no	$\|\Phi^+\rangle\|+\rangle + \|\Phi^-\rangle\|-\rangle$		GHZ state
8	3		2^1	no	no	$\|+\rangle^{\otimes 3} \pm \|-\rangle^{\otimes 3}$		
9	3		2^0	yes	no	$\|+\rangle^{\otimes 3} + \|-\rangle^{\otimes 3}$		

Table A.1 (Color online) : TCM spin systems with $1 \leq N \leq 3$ spins. γ is the degeneracy, **AE** states if **A**nyonic **E**xcitations are possible and **NL** if **N**on **L**ocal measurements are necessary to determine topological quantum numbers v_{top}. The column **GS** lists the **G**round **S**tates up to a normalizing constant $\mathcal{N}$. **LU/S** denotes the equivalence class with respect to **L**ocal **U**nitaries and qubit **S**wap operations in a graph state representation.

[39] Note that this definition of *local* and *nonlocal* operators does not coincide with the common notion of locality used in the theory of entanglement.

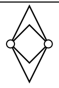
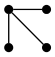
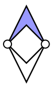
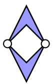
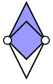
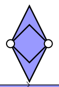
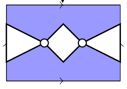
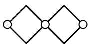
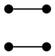
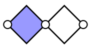
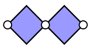
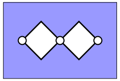
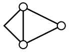
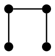
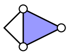
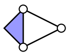
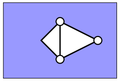
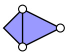
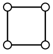
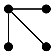
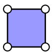
No.	N	**Setting**	γ	**AE**	**NL**	**GS** $\mathcal{N}^{-1}\left\|\mathrm{GS}_i\right\rangle$, $1 \leq i \leq \gamma$	**LU/S**	**Comment**
10	4		2^3	no	no	$\|\Psi^+\rangle\|\Phi^+\rangle \pm \|\Psi^-\rangle\|\Phi^-\rangle$, $\|\Psi^+\rangle\|\Psi^+\rangle \pm \|\Psi^-\rangle\|\Psi^-\rangle$, $\|\Phi^+\rangle\|\Phi^+\rangle \pm \|\Phi^-\rangle\|\Phi^-\rangle$, $\|\Phi^+\rangle\|\Psi^+\rangle \pm \|\Phi^-\rangle\|\Psi^-\rangle$		GHZ state
11	4		2^2	(yes)	no	$\|\Phi^+\rangle\|\Phi^+\rangle \pm \|\Phi^-\rangle\|\Phi^-\rangle$, $\|\Phi^+\rangle\|\Psi^+\rangle \pm \|\Phi^-\rangle\|\Psi^-\rangle$		
12	4		2^1	(yes)	no	$\|\Phi^+\rangle\|\Phi^+\rangle \pm \|\Phi^-\rangle\|\Phi^-\rangle$		
13	4		2^1	(yes)	no	$\|\Phi^+\rangle\|\Phi^+\rangle + \|\Phi^-\rangle\|\Phi^-\rangle$, $\|\Phi^+\rangle\|\Psi^+\rangle + \|\Phi^-\rangle\|\Psi^-\rangle$		
14	4		2^0	(yes)	no	$\|\mathrm{GHZ}_4\rangle$		
15	4		2^2	(yes)	no	$\|\Psi^+\rangle\|\Psi^+\rangle \pm \|\Psi^-\rangle\|\Psi^-\rangle$, $\|\Phi^+\rangle\|\Phi^+\rangle \pm \|\Phi^-\rangle\|\Phi^-\rangle$		non-planar
16	4		2^2	no	no	$\|\Phi^+\rangle\|\Phi^+\rangle$,$\|\Phi^+\rangle\|\Psi^+\rangle$, $\|\Psi^+\rangle\|\Phi^+\rangle$,$\|\Psi^+\rangle\|\Psi^+\rangle$		biseparable
17	4		2^1	(yes)	no	$\|\Phi^+\rangle\|\Psi^+\rangle$,$\|\Phi^+\rangle\|\Phi^+\rangle$		
18	4		2^0	(yes)	no	$\|\Phi^+\rangle\|\Phi^+\rangle$		
19	4		2^1	yes	no	$\|\Phi^+\rangle\|\Phi^+\rangle$,$\|\Psi^+\rangle\|\Psi^+\rangle$		
20	4		2^2	no	no	$\|\Phi^+\rangle\|+\rangle^{\otimes 2} \pm \|\Phi^-\rangle\|-\rangle^{\otimes 2}$, $\|\Psi^+\rangle\|+\rangle^{\otimes 2} \pm \|\Psi^-\rangle\|-\rangle^{\otimes 2}$		
21	4		2^1	yes	no	$\|\Phi^+\rangle\|+\rangle^{\otimes 2} + \|\Phi^-\rangle\|-\rangle^{\otimes 2}$, $\|\Psi^+\rangle\|+\rangle^{\otimes 2} + \|\Psi^-\rangle\|-\rangle^{\otimes 2}$		
22	4		2^1	(yes)	no	$\|\Phi^+\rangle\|+\rangle^{\otimes 2} \pm \|\Phi^-\rangle\|-\rangle^{\otimes 2}$		
23	4		2^1	yes	no	$\|\Phi^+\rangle\|+\rangle^{\otimes 2} + \|\Phi^-\rangle\|-\rangle^{\otimes 2}$, $\|\Psi^+\rangle\|+\rangle^{\otimes 2} - \|\Psi^-\rangle\|-\rangle^{\otimes 2}$		
24	4		2^0	yes	no	$\|\Phi^+\rangle\|+\rangle^{\otimes 2} + \|\Phi^-\rangle\|-\rangle^{\otimes 2}$		
25	4		2^1	no	yes	$\|+\rangle^{\otimes 4} \pm \|-\rangle^{\otimes 4}$		GHZ state
26	4		2^0	yes	no	$\|+\rangle^{\otimes 4} + \|-\rangle^{\otimes 4}$		

TABLE A.2 (Color online) : TCM spin systems with $N = 4$ spins. γ is the degeneracy, **AE** states if **A**nyonic **E**xcitations are possible and **NL** if **N**on **L**ocal measurements are necessary to determine topological quantum numbers v_{top}. The column **GS** lists the **G**round **S**tates up to a normalizing constant $\mathcal{N}$. **LU/S** denotes the equivalence class with respect to **L**ocal **U**nitaries and qubit **S**wap operations in a graph state representation.

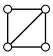
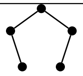
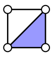
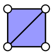
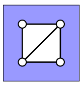
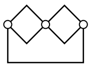
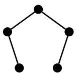
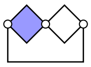
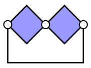
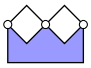
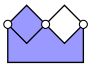
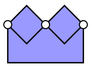
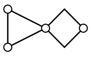
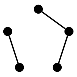
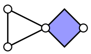
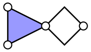
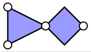
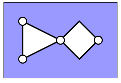
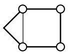
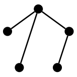
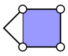
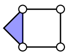
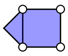
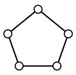
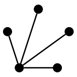
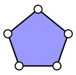
No.	N	**Setting**	γ	**AE**	**NL**	**GS** $\mathcal{N}^{-1}\,\lvert\mathrm{GS}_i\rangle,\ 1\leq i\leq\gamma$	**LU/S**	**Comment**
27	5		2^2	no	no	$\lvert\square\rangle_\pm\lvert\circ\!-\!\circ\rangle+\left[\lvert+\rangle^{\otimes2}\lvert-\rangle^{\otimes2}\pm\lvert-\rangle^{\otimes2}\lvert+\rangle^{\otimes2}\right]\lvert-\rangle,$ $\lvert\square\rangle_\pm\lvert\circ\!-\!\circ\rangle-\left[\lvert+\rangle^{\otimes2}\lvert-\rangle^{\otimes2}\pm\lvert-\rangle^{\otimes2}\lvert+\rangle^{\otimes2}\right]\lvert-\rangle$		
28	5		2^1	yes	no	$\lvert\square\rangle_\pm\lvert\circ\!-\!\circ\rangle\pm\left[\lvert+\rangle^{\otimes2}\lvert-\rangle^{\otimes2}\pm\lvert-\rangle^{\otimes2}\lvert+\rangle^{\otimes2}\right]\lvert-\rangle$		
29	5		2^0	yes	no	$\lvert\boxdot\rangle\lvert\circ\!-\!\circ\rangle+\left[\lvert+\rangle^{\otimes2}\lvert-\rangle^{\otimes2}+\lvert-\rangle^{\otimes2}\lvert+\rangle^{\otimes2}\right]\lvert-\rangle$		
30	5		2^1	yes	no	$\lvert\boxdot\rangle\lvert\circ\!-\!\circ\rangle\pm\left[\lvert+\rangle^{\otimes2}\lvert-\rangle^{\otimes2}+\lvert-\rangle^{\otimes2}\lvert+\rangle^{\otimes2}\right]\lvert-\rangle$		
31	5		2^3	no	no	$\lvert A^+\rangle\lvert B^+\rangle\lvert+\rangle\pm\lvert A^-\rangle\lvert B^-\rangle\lvert-\rangle$ where $A,B\in\{\Psi,\Phi\}$		
32	5		2^2	(yes)	no	$\lvert\Phi^+\rangle\lvert B^+\rangle\lvert+\rangle\pm\lvert\Phi^-\rangle\lvert B^-\rangle\lvert-\rangle$ where $B\in\{\Psi,\Phi\}$		
33	5		2^1	(yes)	no	$\lvert\Phi^+\rangle\lvert\Phi^+\rangle\lvert+\rangle\pm\lvert\Phi^-\rangle\lvert\Phi^-\rangle\lvert-\rangle$		
34	5		2^2	yes	no	$\lvert A^+\rangle\lvert B^+\rangle\lvert+\rangle+\lvert A^-\rangle\lvert B^-\rangle\lvert-\rangle$ where $A,B\in\{\Psi,\Phi\}$		
35	5		2^1	yes	no	$\lvert\Phi^+\rangle\lvert B^+\rangle\lvert+\rangle+\lvert\Phi^-\rangle\lvert B^-\rangle\lvert-\rangle$ where $B\in\{\Psi,\Phi\}$		
36	5		2^0	yes	no	$\lvert\Phi^+\rangle\lvert\Phi^+\rangle\lvert+\rangle+\lvert\Phi^-\rangle\lvert\Phi^-\rangle\lvert-\rangle$		
37	5		2^2	no	no	$\lvert\triangleright\rangle_+\lvert\Psi^+\rangle,\lvert\triangleright\rangle_-\lvert\Psi^+\rangle,$ $\lvert\triangleright\rangle_+\lvert\Phi^+\rangle,\lvert\triangleright\rangle_-\lvert\Phi^+\rangle$		biseparable
38	5		2^1	(yes)	no	$\lvert\triangleright\rangle_+\lvert\Phi^+\rangle,\lvert\triangleright\rangle_-\lvert\Phi^+\rangle$		
39	5		2^1	yes	no	$\lvert\triangleright\rangle_+\lvert\Phi^+\rangle,\lvert\triangleright\rangle_+\lvert\Psi^+\rangle$		
40	5		2^0	yes	no	$\lvert\blacktriangleright\rangle\lvert\Phi^+\rangle$		
41	5		2^1	yes	no	$\lvert\triangleright\rangle_+\lvert\Phi^+\rangle,\lvert\triangleright\rangle_-\lvert\Psi^+\rangle$		
42	5		2^2	no	yes	$\lvert\square\rangle_\pm\lvert\circ\!-\!\circ\rangle+\left[\lvert-\rangle\lvert+\rangle^{\otimes3}\pm\lvert+\rangle\lvert-\rangle^{\otimes3}\right]\lvert-\rangle,$ $\lvert\square\rangle_\pm\lvert\circ\!-\!\circ\rangle-\left[\lvert-\rangle\lvert+\rangle^{\otimes3}\pm\lvert+\rangle\lvert-\rangle^{\otimes3}\right]\lvert-\rangle$		
43	5		2^1	yes	no	$\lvert\boxdot\rangle\lvert\circ\!-\!\circ\rangle\pm\left[\lvert-\rangle\lvert+\rangle^{\otimes3}+\lvert+\rangle\lvert-\rangle^{\otimes3}\right]\lvert-\rangle$		
44	5		2^1	(yes)	yes	$\lvert\square\rangle_\pm\lvert\circ\!-\!\circ\rangle+\left[\lvert-\rangle\lvert+\rangle^{\otimes3}\pm\lvert+\rangle\lvert-\rangle^{\otimes3}\right]\lvert-\rangle$		
45	5		2^0	yes	no	$\lvert\boxdot\rangle\lvert\circ\!-\!\circ\rangle+\left[\lvert-\rangle\lvert+\rangle^{\otimes3}+\lvert+\rangle\lvert-\rangle^{\otimes3}\right]\lvert-\rangle$		
46	5		2^1	no	yes	$\lvert+\rangle^{\otimes5}\pm\lvert-\rangle^{\otimes5}$		GHZ state
47	5		2^0	yes	no	$\lvert+\rangle^{\otimes5}+\lvert-\rangle^{\otimes5}$		

TABLE A.3 (Color online) : TCM spin systems with $N = 5$ spins. γ is the degeneracy, **AE** states if **A**nyonic **E**xcitations are possible and **NL** if **N**on **L**ocal measurements are necessary to determine topological quantum numbers v_{top}. The column **GS** lists the **G**round **S**tates up to a normalizing constant $\mathcal{N}$. **LU/S** denotes the equivalence class with respect to **L**ocal **U**nitaries and qubit **S**wap operations in a graph state representation.

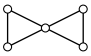
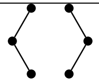
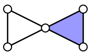
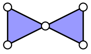
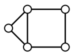
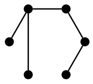
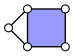
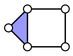
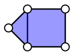
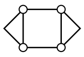
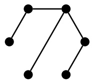
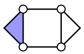
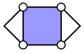
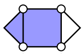
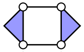
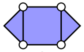
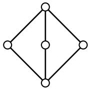
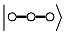
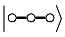
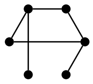
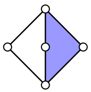
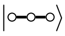
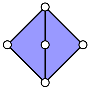
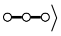
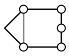
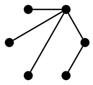
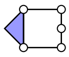
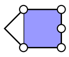
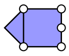
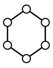
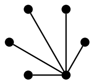
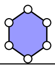
No.	N	**Setting**	γ	**AE**	**NL**	**GS** $\mathcal{N}^{-1}\lvert\mathrm{GS}_i\rangle$, $1 \leq i \leq \gamma$	**LU/S**	**Comment**
48	6		2^2	no	no	$\lvert$ $\rangle_+$ $\lvert$ $\rangle_+$, $\lvert$ $\rangle_+$ $\lvert$ $\rangle_-$, $\lvert$ $\rangle_-$ $\lvert$ $\rangle_+$,$\lvert$ $\rangle_-$ $\lvert$ $\rangle_-$		biseparable
49	6		2^1	yes	no	$\lvert$ $\rangle_+$ $\lvert$ $\rangle$, $\lvert$ $\rangle_-$ $\lvert$ $\rangle$		
50	6		2^0	yes	no	$\lvert$ $\rangle$ $\lvert$ $\rangle$		
51	6		2^2	no	yes	$\lvert$ $\rangle_\pm$ $\lvert$ $\rangle$ $\lvert$ $\rangle$ + $\left[\lvert-\rangle\lvert+\rangle^{\otimes 3} \pm \lvert+\rangle\lvert-\rangle^{\otimes 3}\right]\lvert-\rangle^{\otimes 2}$, $\lvert$ $\rangle_\pm$ $\lvert$ $\rangle$ $\lvert$ $\rangle$ − $\left[\lvert-\rangle\lvert+\rangle^{\otimes 3} \pm \lvert+\rangle\lvert-\rangle^{\otimes 3}\right]\lvert-\rangle^{\otimes 2}$		
52	6		2^1	yes	no	$\lvert$ $\rangle$ $\lvert$ $\rangle$ $\lvert$ $\rangle$ $\pm \left[\lvert-\rangle\lvert+\rangle^{\otimes 3} + \lvert+\rangle\lvert-\rangle^{\otimes 3}\right]\lvert-\rangle^{\otimes 2}$		
53	6		2^1	yes	yes	$\lvert$ $\rangle_\pm$ $\lvert$ $\rangle$ $\lvert$ $\rangle$ + $\left[\lvert-\rangle\lvert+\rangle^{\otimes 3} \pm \lvert+\rangle\lvert-\rangle^{\otimes 3}\right]\lvert-\rangle^{\otimes 2}$		
54	6		2^0	yes	no	$\lvert$ $\rangle$ $\lvert$ $\rangle$ $\lvert$ $\rangle$ + $\left[\lvert-\rangle\lvert+\rangle^{\otimes 3} + \lvert+\rangle\lvert-\rangle^{\otimes 3}\right]\lvert-\rangle^{\otimes 2}$		
55	6		2^3	no	yes	$\lvert A^+\rangle\lvert+\rangle^{\otimes 2}\lvert B^+\rangle \pm \lvert A^-\rangle\lvert-\rangle^{\otimes 2}\lvert B^-\rangle$; $A, B \in \{\Phi, \Psi\}$		
56	6		2^2	(yes)	yes	$\lvert\Phi^+\rangle\lvert+\rangle^{\otimes 2}\lvert B^+\rangle \pm \lvert\Phi^-\rangle\lvert-\rangle^{\otimes 2}\lvert B^-\rangle$; $B \in \{\Phi, \Psi\}$		
57	6		2^2	yes	no	$\lvert A^+\rangle\lvert+\rangle^{\otimes 2}\lvert B^+\rangle + \lvert A^-\rangle\lvert-\rangle^{\otimes 2}\lvert B^-\rangle$; $A, B \in \{\Phi, \Psi\}$		
58	6		2^1	yes	no	$\lvert\Phi^+\rangle\lvert+\rangle^{\otimes 2}\lvert B^+\rangle + \lvert\Phi^-\rangle\lvert-\rangle^{\otimes 2}\lvert B^-\rangle$; $B \in \{\Phi, \Psi\}$		
59	6		2^1	(yes)	yes	$\lvert\Phi^+\rangle\lvert+\rangle^{\otimes 2}\lvert\Phi^+\rangle \pm \lvert\Phi^-\rangle\lvert-\rangle^{\otimes 2}\lvert\Phi^-\rangle$		
60	6		2^0	yes	no	$\lvert\Phi^+\rangle\lvert+\rangle^{\otimes 2}\lvert\Phi^+\rangle + \lvert\Phi^-\rangle\lvert-\rangle^{\otimes 2}\lvert\Phi^-\rangle$		
61	6		2^2	no	yes	$\lvert$ $\rangle_\pm$ $\lvert$$\rangle$+$\left[\lvert+\rangle^{\otimes 2}\lvert-\rangle^{\otimes 2} \pm \lvert-\rangle^{\otimes 2}\lvert+\rangle^{\otimes 2}\right]\lvert-\rangle^{\otimes 2}$, $\lvert$ $\rangle_\pm$ $\lvert$$\rangle$ − $\left[\lvert+\rangle^{\otimes 2}\lvert-\rangle^{\otimes 2} \pm \lvert-\rangle^{\otimes 2}\lvert+\rangle^{\otimes 2}\right]\lvert-\rangle^{\otimes 2}$		
62	6		2^1	yes	yes	$\lvert$ $\rangle_\pm$ $\lvert$$\rangle$ + $\left[\lvert+\rangle^{\otimes 2}\lvert-\rangle^{\otimes 2} \pm \lvert-\rangle^{\otimes 2}\lvert+\rangle^{\otimes 2}\right]\lvert-\rangle^{\otimes 2}$		
63	6		2^0	yes	no	$\lvert$ $\rangle$ $\lvert$$\rangle$ + $\left[\lvert+\rangle^{\otimes 2}\lvert-\rangle^{\otimes 2} + \lvert-\rangle^{\otimes 2}\lvert+\rangle^{\otimes 2}\right]\lvert-\rangle^{\otimes 2}$		
64	6		2^2	no	yes	$\lvert A^+\rangle$ $\lvert$ $\rangle^{\otimes 4} \pm \lvert A^-\rangle\lvert-\rangle^{\otimes 4}$; $A \in \{\Phi, \Psi\}$		
65	6		2^1	(yes)	yes	$\lvert\Phi^+\rangle$ $\lvert$ $\rangle^{\otimes 4} \pm \lvert\Phi^-\rangle\lvert-\rangle^{\otimes 4}$		
66	6		2^1	yes	no	$\lvert A^+\rangle$ $\lvert$ $\rangle^{\otimes 4} + \lvert A^-\rangle\lvert-\rangle^{\otimes 4}$; $A \in \{\Phi, \Psi\}$		
67	6		2^0	yes	no	$\lvert\Phi^+\rangle$ $\lvert$ $\rangle^{\otimes 4} + \lvert\Phi^-\rangle\lvert-\rangle^{\otimes 4}$		
68	6		2^1	no	yes	$\lvert+\rangle^{\otimes 6} \pm \lvert-\rangle^{\otimes 6}$		GHZ state
69	6		2^0	yes	no	$\lvert+\rangle^{\otimes 6} + \lvert-\rangle^{\otimes 6}$		

TABLE A.4 (Color online) : TCM spin systems with $N = 6$ spins. γ is the degeneracy, **AE** states if **A**nyonic **E**xcitations are possible and **NL** if **N**on **L**ocal measurements are necessary to determine topological quantum numbers v_{top}. The column **GS** lists the **G**round **S**tates up to a normalizing constant $\mathcal{N}$. **LU/S** denotes the equivalence class with respect to **L**ocal **U**nitaries and qubit **S**wap operations in a graph state representation.

Appendix B

Computation of LC-orbits

The following two sections outline the MATHEMATICA-scripts used for the computation of LC-classes. The mathematical framework used here is described in sec. 4.2. There are two sections: The first covers the computation of LC-classes and the subsequent analysis concerning graph isomorphisms in order to compute some of the results published in ref. [2] (Table II.). The second outlines the script used in section 4.3 to probe larger systems ($N \gtrsim 10$) for local graph states.

B.1 Size of LC-classes and 2-colorability

In order to compute the whole LC-class of a graph defined by an adjacency matrix G, simultaneously probing for local graphs defined by the locality matrix LOC, the following preliminaries are necessary:

First, the dimension Nu of the system is determined and the null matrix ZERO is defined. Furthermore the complementation of the locality matrix is computed and labeled as CLOC:

```
(* Size of the system *)
Nu=Dimensions[G][[1]];

(* Null matrix *)
ZERO=ConstantArray[0,{Nu,Nu}];

(* Complement locality matrix LOC *)
CLOC=Mod[ConstantArray[1,{Nu,Nu}]+LOC,2];
```

In addition two functions are defined that will be used later on to compute the matrices MA, MB, MC and MD efficiently:

```
(* Switch modulo 3 *)
f3[step_,el_]:=Switch[step[[el]],
  0,{0,1},
  1,{1,0},
  2,{1,1}
];

(* Switch modulo 2 depending on modulo 3 switch *)
f2[step_,el_,type_]:=Switch[type,
  0,Switch[step[[el]],0,{0,1},1,{1,1}],
  1,Switch[step[[el]],0,{1,0},1,{1,1}],
  2,Switch[step[[el]],0,{1,0},1,{0,1}]
];
```

The following three blocks of code comprise the main loop that computes every LC-equivalent graph of G. First, some variables and empty lists are defined. Furthermore the matrices MA, MB, MC and MD are initialised as null matrices. The For-loop runs over the 3^N possible combinations of MD and MB matrices. The latter are generated using function f3:

```
(* Define variables *)
NS=0;NSS=0;NG=0;DG=0;BP=0;
MA=ZERO;MB=ZERO;MC=ZERO;MD=ZERO;
Graphs={};
LOCX={};

(* Check all possible solutions *)
For[s3=0,s3<3^(Nu),s3++,

  (* Construct B and D matrices *)
  S3=IntegerDigits[s3,3,Nu];
  For[l=1,l<=Nu,l++,
    MD[[l,l]]=f3[S3,l][[1]];
    MB[[l,l]]=f3[S3,l][[2]];
  ];
  BUF=G.MB+MD;
```

For each combination of `MD` and `MB` there are 2^N combinations of `MA` and `MC` matrices. These are computed by the second `For`-loop only if `BUF` turns out to be invertible. At last, if all four matrices `MA`, `MB`, `MC` and `MD` are defined, the solution `X` (a potential adjacency matrix of a LC-equivalent graph) is computed:

```
  (* Proceed only if invertible *)
  If[Det[BUF,Modulus->2]==1,
    For[s2=0,s2<2^(Nu),s2++,

      NS++;

      (* Construct A and C matrices *)
      S2=IntegerDigits[s2,2,Nu];
      For[l=1,l<=Nu,l++,
        MA[[l,l]]=f2[S2,l,S3[[l]]][[1]];
        MC[[l,l]]=f2[S2,l,S3[[l]]][[2]];
      ];

      (* Solve linear system *)
      X=LinearSolve[BUF,-G.MA-MC,Modulus->2];
```

The last block of code analyses the solution `X`. First, it is checked whether `X` is an adjacency matrix (this is not always the case). Subsequently it is checked whether the graph has already been computed (often there are duplicates). If not, the 2-colorability and subsequently the locality is probed (by element-wise multiplication of `X` and `CLOC`, see def. 4.1.1):

```
      (* Check whether it represents a graph *)
      If[X*IdentityMatrix[Nu]==ZERO && Transpose[X]==X,

        NG++;

        (* New? Then append to list, check if 2-colorable *)
        If[!MemberQ[Graphs,X],

          DG++;
          AppendTo[Graphs,X];
          If[BipartiteGraphQ[AdjacencyGraph[X]],BP++;];

          (* Check if graph is local with respect to LOC *)
          If[X*CLOC==ZERO,NSS++;AppendTo[LOCX,X];];
        ];
      ];
    ];
  ];
];

(* Print computed data... *)
```

To compute the non-isomorphic (local) graphs stored in `Graphs` and `LOCX`, the following function returns `True` iff the adjacency matrices `gph1` and `gph1` describe isomorphic graphs:

```
g[gph1_,gph2_]:=IsomorphicGraphQ[AdjacencyGraph[gph1],AdjacencyGraph[gph2]];
```

The following loop stores only non-isomorphic graphs of `Graphs` in `IsoGraph` and non-isomorphic local graphs of `LOCX` in `IsoLOCX`:

```
(* Define variables *)
nonisoGraph=0;nonisoLOCX=0;
IsoGraph={};IsoLOCX={};

(* Find non–isomorphic graphs *)
For[i=1,i<= Length[Graphs],i++,

  isin=False;

  For[j=1,j<=Length[IsoGraph],j++,
     If[g[Graphs[[i]],IsoGraph[[j]]],isin=True; Break[];]
  ];

  If[!isin,nonisoGraph++;AppendTo[IsoGraph,Graphs[[i]]]];
];

(* Print computed data... *)

(* Find non–isomorphic LOCAL graphs *)
For[i=1,i<= Length[LOCX],i++,

  isin=False;

  For[j=1,j<=Length[IsoLOCX],j++,
     If[g[LOCX[[i]],IsoLOCX[[j]]],isin=True; Break[];]
  ];

  If[!isin,nonisoLOCX++;AppendTo[IsoLOCX,LOCX[[i]]]];
];

(* Print computed data... *)
```

As a result the total numbers of non-isomorphic (local) graphs `nonisoGraph` and `nonisoLOCX` are printed. It is now easy to verify the numbers published in [2] (Table II.) denoted by |LUclass| and the property of 2-colorability $2 - \text{col}$[40].

B.2 Existence of local graph states

If one only wants to probe whether there are graphs in the LC-class of `G` that are local with respect to the adjacency relation defined by the locality matrix `LOC`, the algorithm explained above should be optimized since the storage of graphs is not necessary. This optimized code was used to prove the nonlocality of `system 4` described in section 4.3. For there are no variables that are changed by the main loop frequently, the parallelization of the code is advantageous[41].

In fact, the following code computes the elements of the considered LC-class in the same way as described in the previous section. If a solution `X` if found the only check performed is whether it is a local graph with respect to `LOC`:

```
(* Initialise kernels *)
LaunchKernels[];

(* Check all possible solutions *)
ParallelDo[

  MA=ZERO;MB=ZERO;MC=ZERO;MD=ZERO;
```

[40] In the course of these computations two potential mistakes in Table II. in ref. [2] have been found: The classes No. 16 and 17 do *not* possess 2-colorable representatives.

[41] The advantage of multithreaded programs is often narrowed by shared variables that lead to mutual blocking requests.

```

  (* Construct B and D matrices *)
  S3=IntegerDigits[s3,3,Nu];
  For[l=1,l<=Nu,l++,
    MD[[l,l]]=f3[S3,l][[1]];
    MB[[l,l]]=f3[S3,l][[2]];
  ];
  BUF=G.MB+MD;

  (* Proceed only if invertible *)
  If[Det[BUF,Modulus->2]==1,
    For[s2=0,s2<2^(Nu),s2++,

        (* Construct A and C matrices *)
        S2=IntegerDigits[s2,2,Nu];
        For[l=1,l<=Nu,l++,
          MA[[l,l]]=f2[S2,l,S3[[l]]][[1]];
          MC[[l,l]]=f2[S2,l,S3[[l]]][[2]];
        ];

        (* Solve linear system *)
        X=LinearSolve[BUF,-G.MA-MC,Modulus->2];

        (* Check if graph is local with respect to LOC *)
        If[X*CLOC==ZERO&&Transpose[X]==X,
          Print["Found local graph at s3=",s3," and s2=",s2];
          (* Print local graph... *)
        ];
    ];
  ];

,{s3,0,3^(Nu)-1}];
```

If the check `X*CLOC==ZERO` is affirmative the corresponding (local) graph is printed. In this case one may abort the computation since the Toric Code system in question is now known to be local. With this code it is possible to show in about 4 hours[42] that `system 4` (12 spins) is nonlocal.

[42] Using a modern personal computer.

Appendix C

Example: An ε-symmetric LC-class

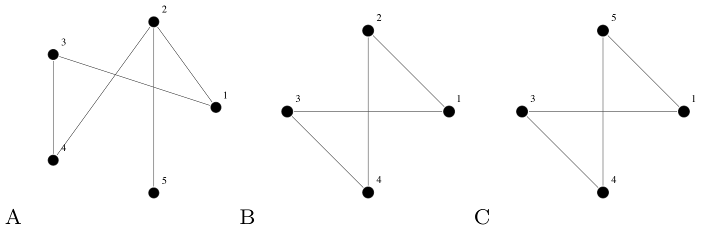


FIGURE C.1 (Color online) : The considered graph G_{52} (A) with inner and outer vertex 2 and 5. The two leaf-reduced graphs $G_{52} - 5$ and $G_{25} - 2$ (B,C) are isomorphic via $2 \leftrightarrow 5$.

To illustrate the structure of ε-symmetric LC-classes used in the proof of Proposition 4.4.3 and depicted in fig. 4.8 consider the leaf-graph G_{52} shown in fig. C.1 A.

Deleting the outer vertex 5 yields the reduced graph $G_{52} - 5$ depicted in fig. C.1 B. Applying $\varepsilon_{52} = \tau_5\tau_2$ to G_{52} permutes 5 and 2 and one obtains G_{25}. Subsequent deletion of the new outer vertex 2 yields $G_{25} - 2$ drawn in fig. C.1 C.

In order to test the statement of Proposition 4.4.3 the whole LC-classes of all three graphs were computed using the `MATHEMATICA`-script described in appendix B. The equivalence classes of $G_{52} - 5$ and $G_{25} - 2$ are depicted in fig. C.2, the class of G_{52} is shown in fig. C.3. In most cases the reader may easily confirm that different graphs of the same LC-class can be transformed into each other by successive application of local complementations.

Let us now probe the statements of subsection 4.4.1:

- The LC-class of G_{52} comprises 30 different graphs (disregarding graph isomorphisms). These 30 graphs are partitioned in four different classes (see fig. C.3). The upper left class A (blue vertices) is characterised by graphs with at least one leaf and outer vertex 5. This is set A in fig. 4.8. Analogously one finds set B in fig. C.3 B (red vertices) that can be obtained from set A by substituting $2 \leftrightarrow 5$. Set C, characterised by an edge between 2 and 5 with symmetric edges to other vertices, is depicted in C (green vertices). One easily verifies that the application of τ_2 and τ_5 yields graphs in A and B, respectively. At last, set D is shown in D (yellow vertices) and may be left by ξ-operations only. In any case, ξ-operations yield graphs in C. Thus the structure depicted in fig. 4.8 is present in this special case.

- The statement of Proposition 4.4.3 is the following: For each graph in fig. C.3 deleting 5 or 2 yields a graph in fig. C.2. More precisely, deleting 5 from graphs in A yields graphs depicted in fig. C.2 A whereas deleting 2 from graphs in B yields graphs in fig. C.2 B. Furthermore deleting 2 *or* 5 from graphs in C *or* D yields graphs in the corresponding class in fig. C.2.

The reader may easily check that these statements hold for all graphs in the LC-class under consideration.

Clearly, every graph in fig. C.3 has *at least* one subgraph in fig. C.2. If one knows that at least one edge of each reduced graph is nonlocal with respect to an adjacency relation regarding the vertices $1, 2, 3, 4$ or $1, 3, 4, 5$ and one finds a 5-spin Toric Code system LC-equivalent to graph states represented by the LC-class of G_{52} such that non-adjacent spins of the reduced systems are non-adjacent in this system as well, it follows the existence of at least one nonlocal edge for each graph state representative in fig. C.3. This is the main idea of Theorem 4.4.6 in a nutshell.

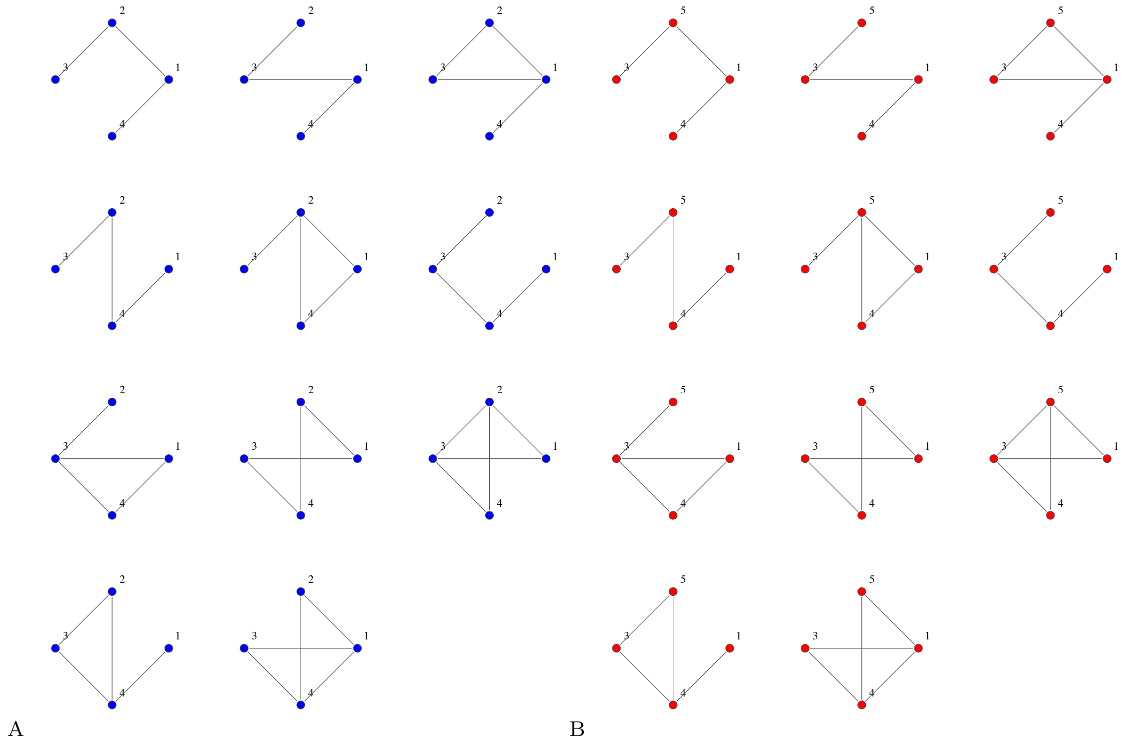


FIGURE C.2 (Color online) : The LC-class obtained from $G_{52} - 5$ is shown in A (blue vertices) whereas the LC-class of $G_{25} - 2$ is depicted in B (red vertices). They are isomorphic via $2 \leftrightarrow 5$ due to the isomorphism connecting $G_{52} - 5$ and $G_{25} - 2$.

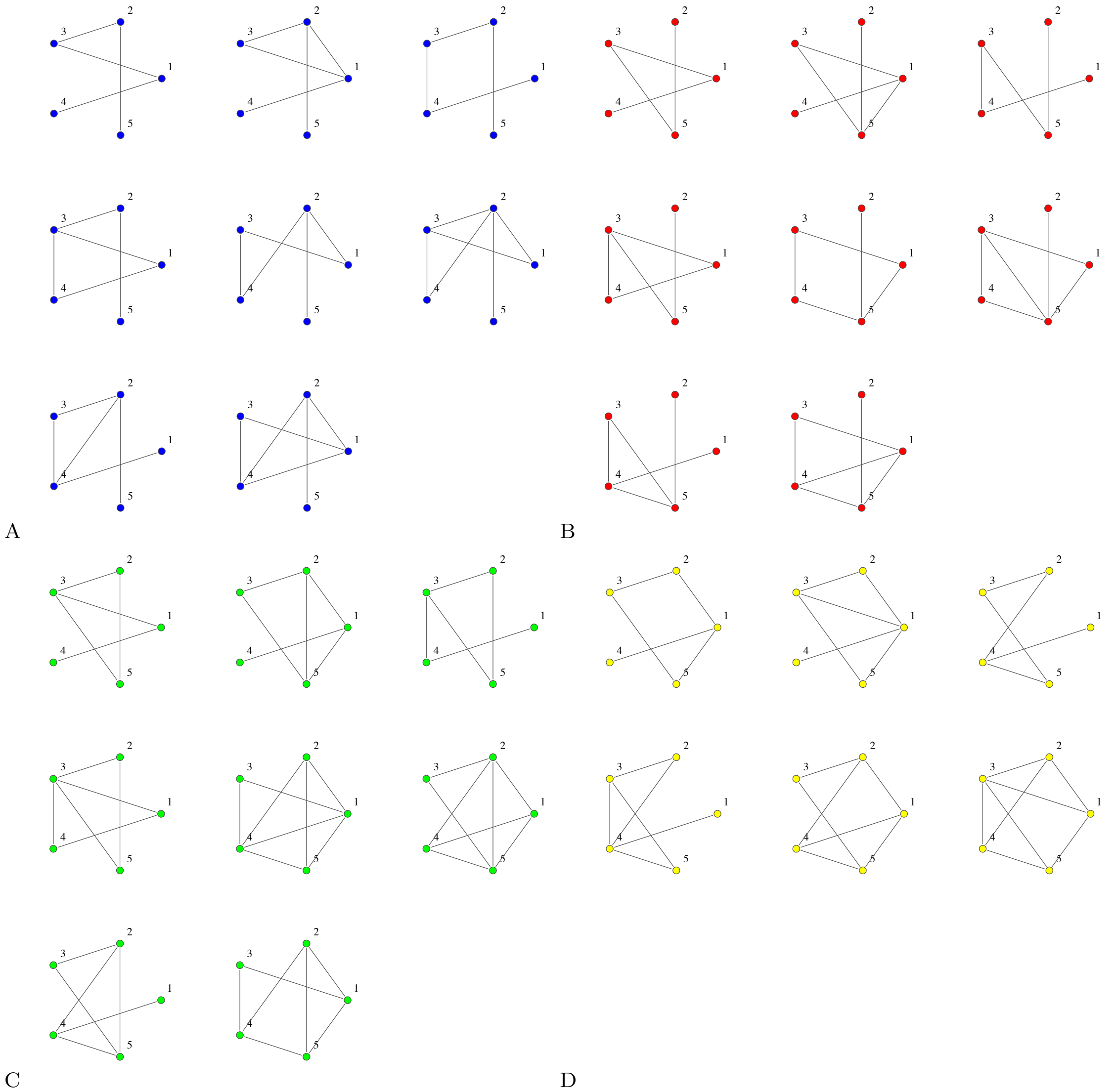


FIGURE C.3 (Color online) : The whole LC-class $[G_{52}]_{\mathrm{LC}} = [G_{25}]_{\mathrm{LC}}$ computed with the MATHEMATICA-script described in appendix B. This class comprises 30 different graphs (disregarding graph isomorphisms) that were partitioned according to the structure depicted in fig. 4.8. There are four partitions: A (blue vertices), B (red vertices), C (green vertices) and D (yellow vertices). See fig. 4.8 for further explanations.

Bibliography

[1] A. Y. Kitaev, Annals of Physics **303**, 2 (2003).

[2] M. Hein, J. Eisert, and H. J. Briegel, Phys. Rev. A **69**, 062311 (2004).

[3] M. Reed and B. Simon, *Methods of Modern Mathematical Physics, Vol. 1: Functional Analysis*, Rev. and enl. ed. (Academic Pr Inc, 1981).

[4] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th anniversary ed. (Cambridge University Press, 2010).

[5] M. Van den Nest, J. Dehaene, and B. De Moor, Phys. Rev. A **69**, 022316 (2004).

[6] D. Deutsch and R. Jozsa, Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences **439**, 553 (1992).

[7] P. W. Shor, SIAM J. Sci. Statist. Comput. **26**, 1484 (1997), quant-ph/9508027.

[8] M. Hein *et al.*, ArXiv Quantum Physics e-prints (2006), arXiv:quant-ph/0602096.

[9] X.-G. Wen, Phys. Rev. Lett. **90**, 016803 (2003).

[10] R. Moessner and S. L. Sondhi, Phys. Rev. Lett. **86**, 1881 (2001).

[11] A. Kitaev, Annals of Physics **321**, 2 (2006), January Special Issue.

[12] D. Arovas, J. R. Schrieffer, and F. Wilczek, Phys. Rev. Lett. **53**, 722 (1984).

[13] S. Trebst, M. Troyer, Z. Wang, and A. W. W. Ludwig, Progress of Theoretical Physics Supplement **176**, 384 (2008), 0902.3275.

[14] J. B. Kogut, Rev. Mod. Phys. **51**, 659 (1979).

[15] H. Weimer, M. Müller, I. Lesanovsky, P. Zoller, and H. P. Büchler, Nature Physics **6**, 382 (2010).

[16] M. Levin and X.-G. Wen, Phys. Rev. Lett. **96**, 110405 (2006).

[17] A. Kitaev and J. Preskill, Phys. Rev. Lett. **96**, 110404 (2006).

[18] Z. Nussinov and G. Ortiz, Annals of Physics **324**, 977 (2009).

[19] J. R. Wootton, ArXiv e-prints (2011), 1103.2878v2.

[20] C. Castelnovo and C. Chamon, Phys. Rev. B **76**, 184442 (2007).

[21] C. Castelnovo and C. Chamon, Phys. Rev. B **78**, 155120 (2008).

[22] Z. Nussinov and G. Ortiz, Phys. Rev. B **77**, 064302 (2008).

[23] W. Fulton, *Algebraic Topology (Graduate Texts in Mathematics)* (Springer, 1995).

[24] R. Diestel, *Graph Theory (Graduate Texts in Mathematics)*, 4th ed. (Springer, 2010).

[25] A. Bouchet, Combinatorica **11**, 315 (1991).

[26] A. Bouchet, Discrete Mathematics **114**, 75 (1993).

[27] P. Sarvepalli and R. Raussendorf, Phys. Rev. A **82**, 022304 (2010).

[28] S. B. Bravyi and A. Y. Kitaev, ArXiv Quantum Physics e-prints (1998), arXiv:quant-ph/9811052.

[29] M. B. Plenio and S. Virmani, Quantum Information & Computation **7**, 1 (2007), quant-ph/0504163.

[30] O. Krueger and R. F. Werner, ArXiv Quantum Physics e-prints (2005), arXiv:quant-ph/0504166.

[31] D. Schlingemann, Quantum Information & Computation **2**, 307 (2002).

[32] M. Van den Nest, J. Dehaene, and B. De Moor, Phys. Rev. A **71**, 062323 (2005).

[33] B. Zeng, H. Chung, A. W. Cross, and I. L. Chuang, Phys. Rev. A **75**, 032325 (2007).

[34] Z. F. Ji, J. X. Chen, Z. H. Wei, and M. S. Ying, Quantum Information & Computation **10**, 97 (2010).

List of Figures