

Deciding Local Unitary Equivalence of Graph States in Quasi-Polynomial Time

Nathan Claudet 

Inria Mocqua and Université de Lorraine, CNRS, LORIA, F-54000 Nancy, France

Simon Perdrix 

Inria Mocqua and Université de Lorraine, CNRS, LORIA, F-54000 Nancy, France

Abstract

We describe an algorithm with quasi-polynomial runtime $n^{\log_2(n)+O(1)}$ for deciding local unitary (LU) equivalence of graph states. The algorithm builds on a recent graphical characterisation of LU-equivalence via generalised local complementation. By first transforming the corresponding graphs into a standard form using usual local complementations, LU-equivalence reduces to the existence of a single generalised local complementation that maps one graph to the other. We crucially demonstrate that this reduces to solving a system of quasi-polynomially many linear equations, avoiding an exponential blow-up. As a byproduct, we generalise Bouchet’s algorithm for deciding local Clifford (LC) equivalence of graph states by allowing the addition of arbitrary linear constraints. We also improve existing bounds on the size of graph states that are LU- but not LC-equivalent. While the smallest known examples involve 27 qubits, and it is established that no such examples exist for up to 8 qubits, we refine this bound by proving that LU- and LC-equivalence coincide for graph states involving up to 19 qubits.

1 Introduction

Graph states form a ubiquitous family of quantum states. They are used as entangled resource states in various quantum information applications, such as measurement-based computation [26, 27, 5], error correction [30, 29, 12, 28], quantum communication network routing [17, 24, 4, 8], and quantum secret sharing [23, 16], to cite a few. In all these applications, graph states are used as multipartite entangled resources, it is thus crucial to understand when two such states have the same entanglement, i.e. when they can be transformed into each other using only local operations. SLOCC-equivalence (stochastic local operations and classical communications) is the most general case that encompasses the use of local unitaries and measurements. In the particular case of graph states, it is enough to consider *LU-equivalence* (local unitaries), as two graph states are SLOCC-equivalent if and only if there exists $U = U_1 \otimes \dots \otimes U_n$ that transforms one state into the other, where each U_i is a single-qubit unitary transformation [19]. One can also consider *LC-equivalence* (local Clifford) which is known to be distinct from LU-equivalence, the smallest known examples of graph states that are LU-equivalent but not LC-equivalent have 27 qubits [21, 33].

As their name suggests, graph states can be uniquely represented by simple undirected graphs. Remarkably, LC-equivalence of graph states is captured by applications of a simple transformation on the corresponding graphs: *local complementation* [14]. Local complementation consists in complementing the neighbourhood of a given vertex. Local complementation was introduced by Kotzig in the 1960s [22], and has been studied independently of its applications in quantum computing. In particular, Bouchet has introduced an efficient algorithm for deciding whether two graphs are related by a sequence of local complementations [1]. This has led to an efficient algorithm for deciding the local Clifford equivalence of graph states within $O(n^4)$ operations, where n is the number of qubits [13].

Recently a graphical characterisation of LU-equivalence has been introduced by means of *generalised local complementation* [11]. The characterisation relies in particular on some

peculiar graph structures called *minimal local sets* that are known to be invariant under local unitary transformations [20]. In [10], it was shown that any vertex is covered by a minimal local set and that a family of minimal local sets covering every vertex of the graph, called an *MLS cover*, can be computed efficiently. Roughly speaking each minimal local set imposes a constraint on the local unitary transformations mapping a graph state to another, so that the existence of such a local unitary is reduced to solving a linear system over integers modulo a power of 2. The solutions can then be graphically interpreted as generalised local complementations.

Shortly after, an algorithm for deciding LU-equivalence was independently introduced [6] based on a similar idea of reducing the problem of LU-equivalence to a linear system, benefiting in particular from the fact that an MLS cover can be computed efficiently. The overall complexity of this algorithm for deciding LU-equivalence depends on two parameters, roughly speaking the size of the linear system and the number of connected components of an intersection graph related to the MLS cover. Both parameters can potentially make the runtime of the algorithm exponential.

We introduce a new algorithm for LU-equivalence of graph states that relies on generalised local complementation and allows us to mitigate both sources of exponential complexity. First, we reduce the LU-equivalence problem to the existence of a single generalised local complementation. To achieve this efficient reduction, we extend Bouchet's algorithm. Then, we demonstrate that the level of the remaining generalised local complementation can be upper bounded by at most the logarithm of the order n of the graphs, leading to a linear system of size at most $n^{\log_2(n)+O(1)}$. This results in an overall algorithm whose time-complexity is quasi-polynomial in n . Notice that the generalisation of Bouchet's algorithm provides an efficient algorithm for deciding whether two graphs are related by local complementations under additional constraints, for instance that the local complementations are applied to a particular subset of vertices.

Thanks to the graphical characterisation of LU-equivalence by means of generalised local complementation, we also address the question of the smallest graphs that are LU- but not LC-equivalent. The study of graph classes where LU-equivalence coincides with LC-equivalence has garnered significant attention [19, 18, 15, 35, 21, 7, 34, 11, 6]. Notably, the smallest known examples of graphs that are LU- but not LC-equivalent have 27 vertices while it is established that no such counterexamples exist for fewer than 8 vertices [7]. We significantly improve this result by showing that any counterexample has at least 20 vertices.

2 Preliminaries

Notations. Given an undirected simple graph $G = (V, E)$, we use the notation $u \sim_G v$ when the vertices u and v are connected in G , i.e. $(u, v) \in E$. $N_G(u) = \{v \in V \mid u \sim_G v\}$ is the neighbourhood of u , $Odd_G(D) = \{v \in V \mid |N_G(v) \cap D| = 1 \bmod 2\}$ is the odd-neighbourhood of the set $D \subseteq V$ of vertices, and $\Lambda_G^D = \{v \in V \mid \forall u \in D, u \sim_G v\}$ is the common neighbourhood of $D \subseteq V$. We assume V totally ordered by a relation \prec . A local complementation with respect to a given vertex u consists in complementing the subgraph induced by the neighbourhood of u , leading to the graph $G \star u = G \Delta K_{N_G(u)}$ where Δ denotes the symmetric difference on edges and K_A is the complete graph on the vertices of A . With a slight abuse of notation we identify multisets of vertices with their multiplicity function $V \rightarrow \mathbb{N}$ (hence we also identify sets of vertices with their indicator functions $V \rightarrow \{0, 1\}$). We consider sums of multisets: for any vertex u , $(S_1 + S_2)(u) = S_1(u) + S_2(u)$. The support $\text{supp}(S)$ of a multiset S of vertices denotes the set of vertices $u \in V$ such that $S(u) > 0$. S is

said independent if no two vertices of $\text{supp}(S)$ are connected. For any multiset S and set D , $S \bullet \Lambda_G^D$ is the number of vertices of S , counted with their multiplicity, that are neighbours to all vertices of D ; in other words, $S \bullet \Lambda_G^D$ is the number of common neighbours of D in S (\bullet is the scalar product: $A \bullet B = \sum_{u \in V} A(u) \cdot B(u)$, so $S \bullet \Lambda_G^D = \sum_{u \in \Lambda_G^D} S(u)$).

To any simple undirected graph $G = (V, E)$, is associated a quantum state $|G\rangle$, called graph state, defined as

$$|G\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{|G[x]|} |x\rangle$$

where n is the order of G and $|G[x]|$ denotes the number of edges in the subgraph of G induced by x .¹

We are interested in the action of local unitaries on graph states. A local unitary is a tensor product of 1-qubit unitaries like Hadamard $H : |a\rangle \mapsto \frac{|0\rangle + (-1)^a |1\rangle}{\sqrt{2}}$, and Z- and X-rotations that are respectively defined as follows:

$$Z(\alpha) := e^{i\frac{\alpha}{2}} \left(\cos\left(\frac{\alpha}{2}\right) I - i \sin\left(\frac{\alpha}{2}\right) Z \right); X(\alpha) := H Z(\alpha) H = e^{i\frac{\alpha}{2}} \left(\cos\left(\frac{\alpha}{2}\right) I - i \sin\left(\frac{\alpha}{2}\right) X \right)$$

where $X : |a\rangle \mapsto |1-a\rangle$ and $Z : |a\rangle \mapsto (-1)^a |a\rangle$. Any 1-qubit unitary can be decomposed into H and $Z(\alpha)$ rotations, whereas 1-qubit Clifford operators are those generated by H and $Z(\frac{\pi}{2})$. Local complementation (denoted by the operator \star) can be implemented by local Clifford operators:

$$|G \star u\rangle = X_u \left(\frac{\pi}{2} \right) \bigotimes_{v \in N_G(u)} Z_v \left(-\frac{\pi}{2} \right) |G\rangle$$

Conversely, if two graph states are related by local Clifford unitaries, the corresponding graphs are related by local complementations [14]. Thus, we use the term of LC-equivalence to describe both local Clifford equivalent graph states, and graphs related by local complementations (conveniently, LC stands for both *local Clifford* and *local complementation*). Similarly, we say that two graphs are LU-equivalent (resp. LC_r -equivalent) when there is a local unitary (resp. a local unitary generated by H and $Z(\frac{\pi}{2^r})$) transforming the corresponding graph states into each other.

Graph states form a subfamily of the well-known *stabilizer states*, indeed $|G\rangle$ is the fix point of $X_u \bigotimes_{v \in N_G(u)} Z_v$ for any $u \in V$. When analysing the entanglement properties of stabilizer states, it is natural to focus on graph states as every stabilizer state is known to be local Clifford equivalent to a graph state [14]. Moreover, there are efficient procedures to associate with any stabilizer state an LC-equivalent graph state [13], thus the problem of deciding the LU-equivalence of stabilizer states naturally reduces to the LU-equivalence of graph states.

We describe in the next section a recent graphical characterisation of LU- and LC_r -equivalences of graph states based on the so-called generalised local complementation [11].

2.1 Generalised local complementation

We review the definition of generalised local complementation and a few of its basic properties. The reader is referred to [11] for a more detailed introduction. A generalised local complementation is a graph transformation parametrised by an independent (multi)set of vertices S and a positive number r called *level*. Like the usual local complementation, the transformation

¹ With a slight abuse of notation, $x \in \{0,1\}^n$ denotes the subset of vertices $\{u \in V | x_{\iota(u)} = 1\}$, where $\iota : V \rightarrow [0, n-1]$ s.t. $u \prec v \Leftrightarrow \iota(u) < \iota(v)$. ι is unique as V is totally ordered by \prec .

consists in toggling some of the edges of the graph depending on the number of neighbours the endpoint vertices have in common in S . Roughly speaking an r -local complementation toggles an edge if the number of common vertices in S is an odd multiple of 2^{r-1} (an example of 2-local complementation is given in Figure 1). To be a valid r -local complementation, the (multi)set S on which the transformation is applied should be r -incident, i.e. the number of common neighbours in S of any set of at most r vertices should be an appropriate power of two:

► **Definition 1** (r -Incidence). *Given a graph G , a multiset S of vertices is r -incident, if for any $k \in [0, r)$, and any $K \subseteq V \setminus \text{supp}(S)$ of size $k + 2$, their number $S \bullet \Lambda_G^K$ of common neighbours in S is a multiple of $2^{r-k-\delta(k)}$, where δ is the Kronecker delta².*

► **Definition 2** (r -Local Complementation). *Given a graph G and an r -incident independent multiset S , let $G \star^r S$ be the graph defined as*

$$u \sim_{G \star^r S} v \Leftrightarrow (u \sim_G v \oplus S \bullet \Lambda_G^{u,v} = 2^{r-1} \bmod 2^r)$$

Below we recall some basic properties of generalised local complementation.

► **Proposition 3.** [11]

- Generalised local complementations are self inverse: if $G \star^r S$ is valid, then $(G \star^r S) \star^r S = G$.
- The multiplicity in S can be upperbounded by 2^r : if $G \star^r S$ is valid, then $G \star^r S = G \star^r S'$, where, for any vertex u , $S'(u) = S(u) \bmod 2^r$.
- If $G \star^r S_1$ and $G \star^r S_2$ are valid and $S_1 + S_2$ is independent in G , then $G \star^r (S_1 + S_2) = (G \star^r S_1) \star^r S_2$.
- If $G \star^r S$ is valid then $G \star^{r+1} (S + S)$ is valid and induces the same transformation: $G \star^{r+1} (S + S) = G \star^{r+1} S \star^{r+1} S = G \star^r S$.
- If $G \star^r S$ is valid then $G \star^{r-1} S$ is valid (when $r > 1$) and $G \star^{r-1} S = G$.

2.2 1- and 2-local complementation

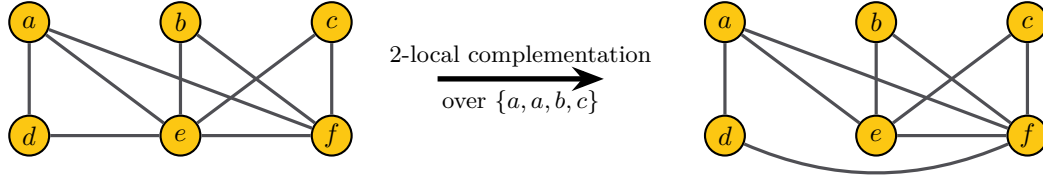
To illustrate how r -local complementation behaves, we consider the simple cases $r = 1$ and $r = 2$. First, any multiset S is 1-incident, and a 1-local complementation is nothing but a sequence of usual local complementations. 2-local complementations cannot always be decomposed into usual local complementations, it is however sufficient to consider 2-local complementations over sets, rather than multisets (see Figure 1):

► **Proposition 4.** *Any 2-local complementation can be decomposed into 1- and 2-local complementations over sets.*

Proof. TOPROVE 0 ◀

The 2-incidence condition can be rephrased as follows when S is a set: for any subset K of $V \setminus S$ of size 2 or 3, there is an even number of common neighbours in S : $|S \cap \Lambda_G^K| = 0 \bmod 2$. In other words, the cut matrix describing the edges between S and $V \setminus S$ is tri-orthogonal [3, 31, 25].

² $\delta(x) \in \{0, 1\}$ and $\delta(x) = 1 \Leftrightarrow x = 0$.



■ **Figure 1** Illustration of a 2-local complementation over the multiset $S = \{a, a, b, c\}$. S is 2-incident: indeed $S \bullet \Lambda_G^{\{d,e,f\}} = 2$, which is a multiple of $2^{2-1-0} = 2$. Similarly, $S \bullet \Lambda_G^{\{d,e\}} = S \bullet \Lambda_G^{\{d,f\}} = 2$ and $S \bullet \Lambda_G^{\{e,f\}} = 4$. Edges de and df are toggled as $S \bullet \Lambda_G^{\{d,e\}} = S \bullet \Lambda_G^{\{d,f\}} = 2 \bmod 4$, but not edge ef as $S \bullet \Lambda_G^{\{e,f\}} = 0 \bmod 4$. Following Proposition 4, the 2-local complementation over S can be decomposed into a 2-local complementation over the set $\{b, c\}$ and a 1-local complementation over the set $\{a\}$.

2.3 LU-equivalence and generalised local complementation

While local complementation can be implemented on graph states by means of local Clifford unitaries, r -local complementations can be implemented on graph states with local unitaries generated by H and $Z(\frac{\pi}{2^r})$:

$$|G \star^r S\rangle = \bigotimes_{u \in V} X\left(\frac{S(u)\pi}{2^r}\right) \bigotimes_{v \in V} Z\left(-\frac{\pi}{2^r} \sum_{u \in N_G(v)} S(u)\right) |G\rangle$$

Conversely, if two graph states are related by local unitaries generated by H and $Z(\frac{\pi}{2^r})$, the corresponding graphs are related by r -local complementations [11]. In other words two graphs are LC_r -equivalent if and only if there is a series of r -local complementations transforming one into another. Two LC_r -equivalent graphs are also LC_{r+1} -equivalent, however the converse does not hold, resulting in an infinite strict hierarchy of local equivalences [11]. Most importantly, generalised local complementations capture the LU-equivalence of graphs:

► **Theorem 5.** [11] *If G_1 and G_2 are LU-equivalent, then G_1 and G_2 are $LC_{\lfloor n/2 \rfloor - 1}$ equivalent, where n is the order of the graphs, i.e. there exists a sequence of $(\lfloor n/2 \rfloor - 1)$ -local complementations transforming G_1 into G_2 .*

Finally, a peculiar property is that for any pair of LU-equivalent graphs, a single generalised complementation, together with usual local complementations, is sufficient to transform one graph into the other:

► **Proposition 6.** [11] *If G_1 and G_2 are LC_r -equivalent, then G_1 and G_2 are related by a sequence of generalised local complementations, such that a single one is of level r , all the others are usual local complementations (i.e. level 1).*

3 Algorithms for LC_r - and LU-equivalences

In this section, we address the problem of deciding whether two given graphs are LU-equivalent. Additionally, we consider a variant of this problem which consists in deciding whether two graphs are LC_r -equivalent for a fixed r . Since LU-equivalent graphs are necessarily LC_r -equivalent for some r , the difference lies in whether the level r is fixed or not.

Thanks to Proposition 6, if G_1 is LC_r -equivalent to G_2 , there exists a single r -local complementation, together with usual local complementations, that transforms G_1 into G_2 . We introduce an algorithm that builds such a sequence of generalised local complementations, in essentially four stages:

- (i) Both G_1 and G_2 are turned in *standard forms* G'_1 and G'_2 by means of (usual) local complementations. These transformations are driven by a so-called minimal local set cover which can be efficiently computed.
- (ii) We then focus on the single r -local complementation: all the possible actions of a single r -local complementation on G'_1 are described as a vector space, for which we compute a basis \mathcal{B} .
- (iii) It remains to find, if it exists, the r -local complementation to apply on G'_1 that leads to G'_2 up to some additional usual local complementations. With an appropriate construction depending on G'_1 , G'_2 and \mathcal{B} , we reduce this problem to deciding whether two graphs are LC-equivalent under some additional requirements on the sequence of local complementations to apply. These requirements can be expressed as linear constraints.
- (iv) Finally, to find such a sequence of local complementations, we apply a variant of Bouchet's algorithm, generalised to accommodate the additional linear constraints.

Stages (i), (iii) and (iv) can be performed in polynomial time in the order n of the graphs. Stage (ii) has essentially a $O(n^r)$ time complexity, thus deciding LC_r -equivalence for a fixed r can be done in polynomial time. Regarding LU-equivalence, Theorem 5 implies $r \leq \frac{n}{2}$. We improve this upperbound and show that r is at most logarithmic in n , leading to a quasi-polynomial time algorithm for LU-equivalence.

The rest of this section is dedicated to the description of the algorithm, its correctness and complexity, beginning with the generalisation of Bouchet's algorithm to decide, in polynomial time, LC-equivalence with additional constraints.

3.1 Bouchet algorithm, revisited

LC-equivalence can be efficiently decided thanks to the famous Bouchet's algorithm [1]. Bouchet proved that LC-equivalence of two given graphs, defined on the same vertex set V , reduces to the existence of subsets of vertices satisfying the following two equations:

► **Proposition 7.** [1] *Two graphs G , G' are LC-equivalent if and only if there exist $A, B, C, D \subseteq V$ such that*

- (i) $\forall u, v \in V$,
 $|B \cap N_G(u) \cap N_{G'}(v)| + |A \cap N_G(u) \cap \{v\}| + |D \cap \{u\} \cap N_{G'}(v)| + |C \cap \{u\} \cap \{v\}| = 0 \pmod{2}$
- (ii) $(A \cap D) \Delta (B \cap C) = V$

While the original proof involves isotropic systems [2], we provide an alternative, self-contained proof in Appendix A, that we believe to be more accessible than the original one.

Notice that Equation (i) is actually a linear equation: the set $\mathcal{S} \subseteq V^4$ of solutions to (i) is a vector space, indeed given two solutions $S = (A, B, C, D)$ and $S' = (A', B', C', D')$ of (i), so is $S + S' = (A \Delta A', B \Delta B', C \Delta C', D \Delta D')$. The linearity of equation (i) can be emphasised using the following encoding:

A set $A \subseteq V$ can be represented by n binary variables $a_1, \dots, a_n \in \mathbb{F}_2$ s.t. $a_v = 1 \Leftrightarrow v \in A$, moreover, with a slight abuse of notations, we identify any set $A \subseteq V$ with the corresponding diagonal \mathbb{F}_2 matrix of dimension $n \times n$ in which diagonal elements are the $(a_v)_{v \in V}$. Following [13, 18], equation (i) is equivalent to

$$\Gamma B \Gamma' + \Gamma A + D \Gamma' + C = 0 \tag{1}$$

and equation (ii) to

$$AD + BC = I \tag{2}$$

where Γ and Γ' are the adjacency matrices of G and G' respectively.

In this section, we consider an extension of Bouchet's algorithm where an additional set of linear constraints on A, B, C and D is added as input of the problem. Such additional linear equations can reflect constraints on the applied local complementations, e.g. deciding whether two graphs are LC-equivalent under the additional constraint that all local complementations are applied on a fixed set V_0 of vertices (see Example 11).

While solving linear equations is easy, equation (ii) is not linear, and the tour de force of Bouchet's algorithm is to point out the fundamental properties of the solutions to both equations (i) and (ii) that allow to decide efficiently the LC-equivalence of graphs. In particular, Bouchet showed that a set of solutions $\mathcal{C} \subseteq \mathcal{S}$ that satisfies both (i) and (ii), is either small or it contains an affine subspace of \mathcal{S} of small co-dimension. In the latter case, the entire set \mathcal{S} is actually an affine space except for some particular cases that can be avoided by assuming that the graphs contain vertices of even degree. We extend this result as follows:

► **Lemma 8.** *Given G, G' two connected graphs with at least one vertex of even degree, and a set L of linear constraints on V^4 , then either the set \mathcal{S}_L of solutions to both L and (i) is of dimension at most 4, or the set $\mathcal{C}_L \subseteq \mathcal{S}_L$ that additionally satisfies (ii) is either empty or an affine subspace of \mathcal{S}_L of codimension at most 2.*

Proof. TOPROVE 1 ◀

► **Remark 9.** Lemma 8 holds actually for any graph that is not in the so-called 'Class α ' of graphs with only odd-degree vertices together with a few additional properties³. When the graphs are in 'Class α ', and in the absence of additional constraints, Bouchet proved that there is at most 2 solutions in \mathcal{C} which do not belong to the affine subspace of small codimension, and these two solutions can be easily computed (see [1], section 7). We leave as an open question the description of the set of solutions for graphs in 'Class α ', in particular when the two particular solutions pointed out by Bouchet do not satisfy L , the set of additional constraints.

From an algorithmic point of view, Lemma 8 leads to a straightforward generalisation of Bouchet's algorithm to efficiently decide LC-equivalence of graphs, under a set of additional linear constraints:

► **Proposition 10.** *Given G, G' two connected graphs of order n with an even-degree vertex, and a set L of ℓ linear constraints on V^4 , one can compute a solution to both (i), (ii) and L when it exists, or claim there is no solution, in runtime $O((n^2 + \ell)n^2)$.*

Proof. TOPROVE 2 ◀

As the algorithm described in Appendix A translates a solution to (i) and (ii), into a sequence of local complementations relating two graphs, some constraints on the sequence of local complementations may be encoded as additional linear constraints. We give a fairly simple example below. A more intricate example is presented in Lemma 21 (in Section 3.3).

► **Example 11.** Let G, G' be two connected graphs of order n with an even-degree vertex, and V_0 a set of vertices. One can decide in runtime $O(n^4)$ whether there exists a sequence of (possibly repeating) vertices $a_1, \dots, a_m \in V_0$ such that $G' = G \star a_1 \star \dots \star a_m$. Roughly

³ (a) any pair of non adjacent vertices should have an even number of common neighbours ; (b) for any cycle C , the number of triangles having an edge in C is equal to the size of C modulo 2.

speaking, the idea is to consider the linear constraint $b_u = 0$ (i.e. $u \in \overline{B}$) for any $u \notin V_0$, to reflect the constraints that local complementations should not be applied outside of V_0 .

An interpretation of the possible additional constraints of the extended Bouchet algorithm in terms of local Clifford operators over graph states is given in Appendix B.

3.2 Minimal local sets and standard form

We consider in this section the first stage of the LU-equivalence algorithm, which consists in putting the two input graphs into a particular shape called standard form by means of local complementations. We adapt a transformation introduced in [11], which is based on the so-called minimal local sets, and turn it into an efficient procedure.

► **Definition 12.** A local set L is a non-empty subset of V of the form $L = D \cup \text{Odd}_G(D)$ for some $D \subseteq V$ called a generator. A minimal local set is a local set that is minimal by inclusion.

A key property of local sets is that they are invariant under LU-equivalence: Two LU-equivalent graphs G_1, G_2 have the same local sets, but not necessarily with the same generators. Moreover, the way the generators of a minimal local set differ in G_1 and G_2 , provides some information on the sequence of generalised local complementations that transforms G_1 into G_2 . It is thus important to cover all vertices of a graph with at least one minimal local set. Fortunately, any graph admits a minimal local set cover (MLS cover for short), and an MLS cover can be computed efficiently, within $O(n^{6.38})$ operations⁴ where n is the order of the graph [10]. The information that an MLS cover provides on each vertex, is reflected by a type X, Y, Z or \perp , defined as follows:

► **Definition 13.** Given a graph G , a vertex u is of type $P \in \{X, Y, Z, \perp\}$ with respect to a MLS cover \mathcal{M} , where P is

- X if for any generator D of a minimal local set of \mathcal{M} containing u , $u \in D \setminus \text{Odd}(D)$,
- Y if for any generator D of a minimal local set of \mathcal{M} containing u , $u \in D \cap \text{Odd}(D)$,
- Z if for any generator D of a minimal local set of \mathcal{M} containing u , $u \in \text{Odd}(D) \setminus D$,
- \perp otherwise.

When a local complementation is applied on a vertex u , its type remains unchanged if it is X or \perp , while types Y and Z are swapped. For the neighbours of u , types Z and \perp remain unchanged, whereas X and Y are exchanged. This leads to a notion of standard form:

► **Definition 14.** A graph G is in standard form with respect to a MLS cover \mathcal{M} if

- There are no vertices of type Y with respect to \mathcal{M} ,
- For every vertex u of type X with respect to \mathcal{M} , any neighbour v of u is of type Z with respect to \mathcal{M} and satisfies $u \prec v$, in particular the vertices of type X with respect to \mathcal{M} form an independent set,
- For every vertex u of type X with respect to \mathcal{M} , $\{u\} \cup N_G(u) \in \mathcal{M}$.

► **Remark 15.** This notion of standard form is a generalisation of the one introduced in [11], where the MLS cover considered consists of every minimal local set of the graph: $\mathcal{M}_{\max} := \{L \subseteq V \mid L \text{ is a minimal local set}\}$. Since there can be exponentially many minimal local sets, using \mathcal{M}_{\max} does not lead to an efficient procedure, for instance when computing the type of each vertex.

⁴ The algorithm presented in [10] computes a MLS cover in $O(n^4)$ evaluations of the so-called cut-rank function, which itself can be computed in $O(n^\omega)$ field operations where $\omega < 2.38$.

Given a pair of LU-equivalent graphs, one can efficiently compute a (common) MLS cover and put both graphs in standard form by means of local complementations:

► **Lemma 16.** *There exists an efficient algorithm that takes as inputs two graphs G_1 and G_2 of order n , and either claim that they are not LU-equivalent, or compute an MLS cover \mathcal{M} and two graphs G'_1 and G'_2 LC-equivalent to G_1 and G_2 respectively, such that G'_1 and G'_2 are both in standard form with respect to \mathcal{M} , in runtime $O(n^{6.38})$.*

The algorithm is fairly similar to the one presented in the proof of Proposition 24 in [11] (the main difference being that we may now add minimal local sets to the MLS cover) and can be found in Appendix C. Notice the most computationally expensive step of the algorithm is the computation of the MLS cover, hence the runtime $O(n^{6.38})$. Standard forms with respect to a common MLS cover implies some strong similarities in the structure of graphs:

► **Lemma 17.** *If two graphs G_1 and G_2 are LU-equivalent and in standard form with respect to an MLS cover \mathcal{M} , then every vertex has the same type in G_1 and G_2 , and every vertex u of type X satisfies $N_{G_1}(u) = N_{G_2}(u)$.*

Lemma 17 was proved in [11] for the maximal MLS cover \mathcal{M}_{\max} , but the mathematical arguments hold for any arbitrary MLS cover. A key argument is that two LU-equivalent graphs have the same vertices of type \perp with respect to any arbitrary MLS cover.

After performing the algorithm described in Lemma 16, one can check in quadratic time⁵ whether each vertex has the same type in G_1 and G_2 , and whether every vertex of type X has the same neighbourhood in both graphs. If either condition is not met, the graphs are not LU-equivalent.

Finally, thanks to standard form, deciding LC_r -equivalence of graphs reduces to determining whether they are related by a single r -local complementation along with some usual local complementations:

► **Lemma 18.** *If G_1 and G_2 are LC_r -equivalent and are both in standard form with respect to an MLS cover \mathcal{M} , then G_1 and G_2 are related by a sequence of local complementations on the vertices of type \perp along with a single r -local complementation over the vertices of type X .*

Lemma 18 was proved in [11] for the maximal MLS cover \mathcal{M}_{\max} , but the mathematical arguments hold for any arbitrary MLS cover.

3.3 An algorithm to recognise LC_r -equivalent graph states

We are now ready to describe the algorithm that recognises two LC_r -equivalent graphs. We consider a level $r \geq 1$, and two graphs G_1 and G_2 of order n , defined on the same vertex set V . Following Lemma 16, assume, without loss of generality, that G_1 and G_2 are both in standard form with respect to the same MLS cover. Then, it is valid (see Lemma 17) to define $V_X, V_Z \subseteq V$, the sets of vertices respectively of type X and Z with respect to the MLS cover. Also, each vertex in V_X has the same neighbourhood in both G_1 and G_2 . According to Lemma 18, if G_1 and G_2 are LC_r -equivalent then there is a single r -local complementation over vertices of V_X together with a series of local complementations on vertices of type \perp that transform G_1 into G_2 . We first focus on the single r -local complementation (that

⁵ In the order of the graphs, assuming the information of the types of the vertices with respect to the MLS cover is conserved.

10:21 Deciding Local Unitary Equivalence of Graph States in Quasi-Polynomial Time

commutes with the local complementations on vertices of type \perp , as there is no edge between a vertex of type X and a vertex of type \perp) and thus consider all the possible graphs that can be reached from G_1 by mean of a single r -local complementation over vertices of V_X . Notice that such a r -local complementation only toggles edges which both endpoints are in V_Z . Given a multiset S , the edges toggled in $G_1 \star^r S$ can be represented by a vector $\omega^{(S)} \in \mathbb{F}_2^{\{u,v \in V_Z \mid u \neq v\}}$ such that for any $u, v \in V_Z$, $\omega_{u,v}^{(S)} = u \sim_{G_1} v \oplus u \sim_{G_1 \star^r S} v$. The actions of all the possible r -local complementations on G_1 can thus be described as the set $\Omega = \{\omega^{(S)} \mid S \text{ is an } r\text{-incident multiset of vertices of type } X\}$.

► **Lemma 19.** Ω is a vector space and a basis \mathcal{B} of Ω can be computed in runtime $O(rn^{r+2.38})$.

Proof. TOPROVE 3 ◀

Thanks to the exhaustive description of all possible r -local complementations on G_1 , we are now ready to reduce LC_r -equivalence to LC -equivalence with some additional constraint. We denote by $G_1^\#$ (resp. $G_2^\#$) the graph obtained from G_1 (resp. G_2) by the following procedure. First, remove the vertices of V_X . Then, for each vector $\omega \in \mathcal{B}$, for each $u, v \in V_Z$ such that $\omega_{u,v} = 1$, add a vertex connected only to u and v and call it $p_{u,v}^\omega$, and let $\mathcal{P}^\omega = \{p_{u,v}^\omega \mid \omega_{u,v} = 1\}$. In the following, we refer to the vertices added by this procedure as "new vertices".

► **Lemma 20.** G_1 and G_2 are LC_r -equivalent if and only if there exists a sequence of (possibly repeating) vertices a_1, \dots, a_m such that $G_2^\# = G_1^\# \star a_1 \star \dots \star a_m$ satisfying the following additional constraints:

- the sequence contains no vertex of V_Z ;
- for each $\omega \in \mathcal{B}$, either the sequence contains every vertex of \mathcal{P}^ω exactly once, or it contains none.

The proof of Lemma 20 makes use of Lemma 18 and is given in Appendix D.

There exists an efficient algorithm that decides whether two graphs are LC -equivalent with such additional constraints using our generalisation of Bouchet's algorithm.

► **Lemma 21.** Deciding whether there exists a sequence of (possibly repeating) vertices a_1, \dots, a_m such that $G_2^\# = G_1^\# \star a_1 \star \dots \star a_m$, satisfying the additional constraints described in Lemma 20, can be done in runtime $O(n^4)$.

Proof. TOPROVE 4 ◀

Summing up, we have an algorithm that decides, for a fixed level r , the LC_r -equivalence of graphs in polynomial runtime.

► **Theorem 22.** There exists an algorithm that decides if two graphs are LC_r -equivalent with runtime $O(rn^{r+2.38} + n^{6.38})$, where n is the order of the graphs.

The algorithm reads as follows:

1. Put G_1 and G_2 in standard form with respect to the same MLS cover if possible, otherwise output NO.
2. Check whether each vertex has the same type in G_1 and G_2 , and whether every vertex of type X has the same neighbourhood in both graphs, otherwise output NO.
3. Compute a basis of the vector space Ω .
4. Compute the graphs $G_1^\#$ and $G_2^\#$.
5. Decide whether $G_1^\#$ and $G_2^\#$ are LC -equivalent with the additional constraints described in Lemma 20. Output YES if this is the case, NO otherwise.

Notice that the algorithm is exponential in r , in particular it does not provide an efficient algorithm to decide LU-equivalence of graph states. To address this issue, we provide in the next subsection some upper bounds on the level of a generalised local complementation.

3.4 Bounds for generalised local complementation

In this section, we prove an upper bound on the level of a valid generalised local complementation: roughly speaking we show that if $G \star^r S$ is valid then r is at most logarithmic in the order of the graph G . This bound is however not true in general as it has been shown in [11] that whenever $G \star^r S$ is valid, we have $G \star^r S = G \star^{r+1} (S + S)$. To avoid these pathological cases, we thus focus on genuine r -incident independent multisets:

► **Definition 23.** *Given a graph G , a r -incident independent multiset S is genuine if there exists a set $K \subseteq V \setminus \text{supp}(S)$ such that $|K| > 1$ and $\sum_{N_G(u)=K} S(u)$ is odd⁶.*

► **Proposition 24.** *If $G \star^r S$ is valid and there is no S' such that $G \star^{r-1} S' = G \star^r S$ then S is a genuine r -incident independent multiset.*

Proof. TOPROVE 5 ◀

Genuine r -incidence can only occur for multisets whose support is of size at least exponential in r .

► **Lemma 25.** *If $r > 1$ and S is a genuine r -incident independent multiset of a graph G , then $|\text{supp}(S)| \geq 2^{r+2} - r - 3$.*

Proof. TOPROVE 6 ◀

Likewise, r -local complementations that cannot be implemented by $(r-1)$ -local complementations can only occur on multisets with sufficiently many vertices outside their support.

► **Lemma 26.** *If $G \star^r S$ is valid and there is no S' such that $G \star^{r-1} S' = G \star^r S$, then $|V \setminus \text{supp}(S)| \geq r + 3$.*

The proof of Lemma 26 involves similar techniques as the proof of Lemma 25 and is given in Appendix E. Lemmas 25 and 26 together give a simple bound involving only the order of the graph.

► **Proposition 27.** *If $G \star^r S$ is valid and there is no S' such that $G \star^{r-1} S' = G \star^r S$, then $n \geq 2^{r+2}$, where n is the order of G .*

Put differently, any r -local complementation on a graph of order at most $2^{r+2} - 1$ can be implemented by $(r-1)$ -local complementations:

► **Corollary 28.** *If two graphs of order at most $2^{r+2} - 1$ are LC_r -equivalent, then they are LC_{r-1} -equivalent.*

In other words, two LC_r -equivalent but not LC_{r-1} -equivalent graphs are of order at least 2^{r+2} . This implies the following strengthening of Theorem 5.

⁶ With a slight abuse of notation, $\sum_{N_G(u)=K} S(u)$ is the sum over all $u \in V$ s.t. $N_G(u) = K$.

► **Corollary 29.** *If two graphs of order at most $2^{r+3} - 1$ are LU-equivalent, they are LC_r -equivalent.*

Proof. TOPROVE 7 ◀

Corollary 29 provides a logarithmic bound on the level of generalised local complementations to consider for LU-equivalence: if two graphs of order $n > 7$ are LU-equivalent then they are $LC_{\lceil \log_2(\frac{n+1}{8}) \rceil}$ -equivalent. This bound leads to a quasi-polynomial time algorithm for LU-equivalence, as described in the next section. Notice that in Section 4, we elaborate on the consequences of Corollary 29 on the minimal order of graphs that are LU- but not LC-equivalent.

3.5 An algorithm to recognise LU-equivalent graph states

According to Theorem 22, we have an algorithm that recognises two LC_r -equivalent graphs of order n in runtime $O(rn^{r+2.38} + n^{6.38})$. According to Corollary 29, G_1 and G_2 are LU-equivalent if and only if they are LC_r -equivalent, where $r = \log_2(n) + O(1)$. Thus, our algorithm that decides LC_r -equivalence translates directly to an algorithm that decides LU-equivalence.

► **Theorem 30.** *There exists an algorithm that decides if two graphs are LU-equivalent with runtime $n^{\log_2(n)+O(1)}$, where n is the order of the graphs.*

In comparison, Burchardt et al. algorithm for LU-equivalence [6] has two sources of exponential time complexity. The logarithmic upper bound on the level of generalised local complementation we introduce may mitigate one of these sources (making one parameter of the complexity quasi-polynomial), but does not affect a priori the second one, which is roughly speaking the number of connected components of an intersection graph related to the MLS cover.

4 LU- and LC-equivalence coincide for graph states up to 19 qubits

It is known that there exists a pair of 27-vertex graphs that are not LC-equivalent, but LU-equivalent, more precisely they are LC_2 -equivalent [21, 33]. It is still an open question whether this is a minimal example (in number of vertices). In other words, does a pair of graphs that are LU-equivalent but not LC-equivalent on 26 vertices or less exist? In theory, one could check every pair of graphs of order up to 26, but the rapid combinatorial explosion in the number of graphs as the number of vertices increases, makes it unfeasible in practice.

The best bound known so far⁷ is that for graphs of order up to 8, $LU=LC$ i.e. LU- and LC-equivalence coincide [7]. The results of Section 3.4 (see Corollary 29) already imply a substantial improvement on this bound: $LU=LC$ for graphs of order up to 15. Furthermore, for graphs of order up to 31, $LU=LC_2$, i.e. if two graphs of order up to 31 are LU-equivalent, they are LC_2 -equivalent. Thus, asking whether $LU=LC$ holds for graphs of order up to 26 is equivalent to asking whether $LC_2=LC$ holds for graphs of order up to 26. One direction is to study when a 2-local complementation on a multiset S can be implemented using only usual local complementations over vertices in the support of S . If this were to be the case for every graph of order up to 26, it would show that the 27-vertex counterexample is minimal

⁷ In [6] it is proved that the number of LU- and LC-orbits of **unlabelled** graphs of order up to 11 is the same.

in number of vertices. In the following we study the structure of 2-local complementation to prove that $LU=LC$ holds for graph of order up to 19.

According to Lemma 26, if there are at most 4 vertices outside the support of some 2-incident independent multiset S , then a 2-local complementation on S can be implemented by usual local complementations. In the peculiar case of 2-local complementation, we are able to use computer-assisted generation (see Appendix F) to extend the result. The code is available at [9].

► **Lemma 31.** *Let S be a 2-incident independent multiset of a graph G . If $|V \setminus \text{supp}(S)| \leq 5$, or if $|V \setminus \text{supp}(S)| = 6$ and $|\text{supp}(S)| \leq 20$, then a 2-local complementation on S can be implemented by local complementations over a subset of $\text{supp}(S)$.*

Likewise, according to Lemma 25 and Proposition 24, if the support of some 2-incident independent multiset S is of size at most 10, then a 2-local complementation on S can be implemented by usual local complementations. To extend this result to 2-incident independent multisets whose supports is of size at most 12, we first study the case of twin-less sets (two distinct non-connected vertices u and v are twins if $N_G(u) = N_G(v)$).

► **Lemma 32.** *Let S be a 2-incident independent set of a graph G such that S does not contain any twins and $|S| \leq 12$. Then, $G \star^2 S = G$.*

The proof of Lemma 32 is an induction over the number of vertices connected to S and is given in Appendix G.

According to Proposition 4, any 2-local complementation can be decomposed into 1- and 2-local complementations over sets. Furthermore, one can check that if an 2-incident independent set S contains two twins u and v , then a 2-local complementation over S has the same effect as a 2-local complementation over $S \setminus \{u, v\}$ followed by a local complementation over u . Thus, the action of a 2-local complementation can be described by a 2-local complementation over a twin-less set followed by usual local complementations. Then, Lemma 32 can be applied on the twin-less set to yield the following result:

► **Lemma 33.** *Let S be a 2-incident independent multiset of a graph G such that $|\text{supp}(S)| \leq 12$. Then, a 2-local complementation over S can be implemented by local complementations over a subset of $\text{supp}(S)$.*

According to Lemma 31 and Lemma 33, if a 2-incident independent multiset S satisfies $|\text{supp}(S)| \leq 12$ or $|V \setminus \text{supp}(S)| \leq 5$, or alternatively if $|\text{supp}(S)| \leq 20$ and $|V \setminus \text{supp}(S)| = 6$, then a 2-local complementation over S can be implemented by usual local complementations. Thus, for graphs of order up to 19, any 2-local complementation can be implemented by usual local complementations, implying $LU=LC$. In other words, a 2-local complementation that cannot be implemented by usual local complementation is possible only on a graph of order at least 20. We summarise our findings in the following proposition:

► **Proposition 34.** *LU - and LC -equivalence coincide for graph states up to 19 qubits.*

5 Conclusion

In this paper, we have introduced a quasi-polynomial runtime algorithm to recognise the LU -equivalence of graph states – and more generally stabilizer states – based on a recent generalisation of local complementation that captures the LU -equivalence of graph states. A key component of our approach is a new, nontrivial logarithmic bound on the level of the generalised local complementation.

We have also extended the well-known Bouchet algorithm to recognise the LC-equivalence of graph states, by allowing the addition of arbitrary linear constraints. This extension enables finer control over local complementations (or local Clifford operators) in the LC-equivalence problem, and we believe it will have broader applications.

We have also made significant progress in understanding the structure of quantum entanglement by demonstrating that LC-equivalence and LU-equivalence coincide for graph states with up to 19 qubits, extending the previously known bound of 8 qubits. The smallest known example of a pair of graph states that are LU- but not LC-equivalent consists of 27 qubits. A natural next step is to determine whether LU- and LC-equivalence continue to coincide for graph states up to 26 qubits or, alternatively, to find a counterexample in the range of 20 to 26 qubits. As shown in this work, leveraging generalised local complementation should facilitate this exploration.

Acknowledgements

The authors want to thank Adam Burchardt for fruitful discussions. This work is supported by the the *Plan France 2030* through the PEPR integrated project EPiQ ANR-22-PETQ-0007 and the HQI platform ANR-22-PNCQ-0002; and by the European projects Quantum Flagship NEASQC, European High-Performance Computing HPCQS and MSCA Staff Exchanges Qcomical HORIZON-MSCA-2023-SE-01. The project is also supported by the *Maison du Quantique* MaQuEst.

References

- 1 André Bouchet. An efficient algorithm to recognize locally equivalent graphs. *Combinatorica*, 11(4):315–329, Dec 1991. doi:10.1007/BF01275668.
- 2 André Bouchet. Isotropic systems. *European Journal of Combinatorics*, 8(3):231–244, 1987. URL: <https://www.sciencedirect.com/science/article/pii/S0195669887800276>, doi:10.1016/S0195-6698(87)80027-6.
- 3 Sergey Bravyi and Jeongwan Haah. Magic-state distillation with low overhead. *Physical Review A—Atomic, Molecular, and Optical Physics*, 86(5):052329, 2012. arXiv:1209.2426, doi:10.1103/PhysRevA.86.052329.
- 4 Sergey Bravyi, Yash Sharma, Mario Szegedy, and Ronald de Wolf. Generating k epr-pairs from an n -party resource state. *Quantum*, 8:1348, 2024. arXiv:2211.06497, doi:10.22331/q-2024-05-14-1348.
- 5 Hans J. Briegel, David E. Browne, Wolfgang Dür, Robert Raussendorf, and Maarten Van den Nest. Measurement-based quantum computation. *Nature Physics*, 5(1):19–26, 2009. arXiv:0910.1116, doi:10.1038/nphys1157.
- 6 Adam Burchardt, Jarn de Jong, and Lina Vandr . Algorithm to verify local equivalence of stabilizer states, 2024. arXiv:2410.03961.
- 7 Ad n Cabello, Antonio J. L pez-Tarrida, Pilar Moreno, and Jos  R. Portillo. Entanglement in eight-qubit graph states. *Physics Letters A*, 373(26):2219–2225, 2009. arXiv:0812.4625, doi:10.1016/j.physleta.2009.04.055.
- 8 Maxime Cautr s, Nathan Claudet, Mehdi Mhalla, Simon Perdrix, Valentin Savin, and St phan Thomass . Vertex-minor universal graphs for generating entangled quantum subsystems. In *Proceedings of the 51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*, 2024. arXiv:2402.06260, doi:10.4230/LIPIcs.ICALP.2024.36.
- 9 Nathan Claudet. LU=LC up to 19 qubits: github.com/nathanclaudet/LU-equals-LC-up-to-19qubits, 2024.

- 10 Nathan Claudet and Simon Perdrix. Covering a graph with minimal local sets. In *Proceedings of the 24th workshop on Graph Theory (WG 2024)*, 2024. [arXiv:2402.10678](#), [doi:10.1007/978-3-031-75409-8_10](#).
- 11 Nathan Claudet and Simon Perdrix. Local equivalence of stabilizer states: a graphical characterisation. In *Proceedings of the 42nd Symposium on Theoretical Aspects of Computer Science (STACS 2025)*, 2025. [arXiv:2409.20183](#), [doi:10.4230/LIPIcs.STACS.2025.27](#).
- 12 Andrew Cross, Graeme Smith, John A. Smolin, and Bei Zeng. Codeword stabilized quantum codes. In *2008 IEEE International Symposium on Information Theory*, pages 364–368. IEEE, 2008. [arXiv:0708.1021](#), [doi:10.1109/TIT.2008.2008136](#).
- 13 Maarten Van den Nest, Jeroen Dehaene, and Bart De Moor. Efficient algorithm to recognize the local Clifford equivalence of graph states. *Phys. Rev. A*, 70:034302, Sep 2004. [arXiv:quant-ph/0405023](#), [doi:10.1103/PhysRevA.70.034302](#).
- 14 Maarten Van den Nest, Jeroen Dehaene, and Bart De Moor. Graphical description of the action of local Clifford transformations on graph states. *Physical Review A*, 69(2), Feb 2004. [arXiv:quant-ph/0308151](#), [doi:10.1103/physreva.69.022316](#).
- 15 Maarten Van den Nest, Jeroen Dehaene, and Bart De Moor. Local unitary versus local Clifford equivalence of stabilizer states. *Physical Review A*, 71(6), Jun 2005. [arXiv:quant-ph/0411115](#), [doi:10.1103/physreva.71.062323](#).
- 16 Sylvain Gravier, Jérôme Javelle, Mehdi Mhalla, and Simon Perdrix. Quantum secret sharing with graph states. In *Proceedings of the 8th Mathematical and Engineering Methods in Computer Science International Doctoral Workshop (MEMICS 2012)*, 2013. URL: <https://hal.science/hal-00933722/document>, [doi:10.1007/978-3-642-36046-6_3](#).
- 17 Frederik Hahn, Anna Pappa, and Jens Eisert. Quantum network routing and local complementation. *npj Quantum Information*, 5(1):1–7, 2019. [arXiv:1805.04559](#), [doi:10.1038/s41534-019-0191-6](#).
- 18 Mayan Hein, Wolfgang Dür, Jens Eisert, Robert Raussendorf, Maarten Van den Nest, and Hans J. Briegel. Entanglement in graph states and its applications. *Quantum computers, algorithms and chaos*, 162, Mar 2006. [arXiv:quant-ph/0602096](#), [doi:10.3254/978-1-61499-018-5-115](#).
- 19 Mayan Hein, Jens Eisert, and Hans J. Briegel. Multiparty entanglement in graph states. *Physical Review A*, 69(6), Jun 2004. [arXiv:quant-ph/0307130](#), [doi:10.1103/physreva.69.062311](#).
- 20 Peter Høyer, Mehdi Mhalla, and Simon Perdrix. Resources required for preparing graph states. In *17th International Symposium on Algorithms and Computation (ISAAC 2006)*, volume 4288 of *Lecture Notes in Computer Science*, pages 638 – 649, Dec 2006. URL: <https://hal.archives-ouvertes.fr/hal-01378771>, [doi:10.1007/11940128_64](#).
- 21 Zhengfeng Ji, Jianxin Chen, Zhaohui Wei, and Mingsheng Ying. The LU-LC conjecture is false, 2007. [arXiv:0709.1266](#).
- 22 Anton Kotzig. Eulerian lines in finite 4-valent graphs and their transformations. *Colloquium on Graph Theory Tihany 1966*, pages 219 – 230, 1968.
- 23 Damian Markham and Barry C. Sanders. Graph states for quantum secret sharing. *Physical Review A*, 78(4):042309, 2008. [arXiv:0808.1532](#), [doi:10.1103/PhysRevA.78.042309](#).
- 24 Clément Meignant, Damian Markham, and Frédéric Grosshans. Distributing graph states over arbitrary quantum networks. *Physical Review A*, 100:052333, Nov 2019. [arXiv:1811.05445](#), [doi:10.1103/PhysRevA.100.052333](#).
- 25 Sepehr Nezami and Jeongwan Haah. Classification of small triorthogonal codes. *Phys. Rev. A*, 106:012437, Jul 2022. [arXiv:2107.09684](#), [doi:10.1103/PhysRevA.106.012437](#).
- 26 Robert Raussendorf and Hans J. Briegel. A one-way quantum computer. *Physical review letters*, 86(22):5188, 2001. [doi:10.1103/PhysRevLett.86.5188](#).
- 27 Robert Raussendorf, Daniel E. Browne, and Hans J. Briegel. Measurement-based quantum computation on cluster states. *Physical review A*, 68(2):022312, 2003. [arXiv:quant-ph/0301052](#), [doi:10.1103/PhysRevA.68.022312](#).

- 28 Pradeep Sarvepalli and Robert Raussendorf. Local equivalence of surface code states. In *Proceedings of the 5th Conference on the Theory of Quantum Computation, Communication, and Cryptography (TQC 2010)*, 2011. URL: https://link.springer.com/content/pdf/10.1007/978-3-642-18073-6_5.pdf, doi:10.1007/978-3-642-18073-6_5.
- 29 Dirk Schlingemann. Stabilizer codes can be realized as graph codes, 2001. [arXiv:quant-ph/0111080](#).
- 30 Dirk Schlingemann and Reinhard F. Werner. Quantum error-correcting codes associated with graphs. *Physical Review A*, 65(1):012308, 2001. [arXiv:quant-ph/0012111](#), doi:10.1103/PhysRevA.65.012308.
- 31 Minjia Shi, Haodong Lu, Jon-Lark Kim, and Patrick Solé. Triorthogonal codes and self-dual codes. *Quantum Information Processing*, 23(7):280, Jul 2024. [arXiv:2408.09685](#), doi:10.1007/s11128-024-04485-9.
- 32 Arne Storjohann. *Algorithms for matrix canonical forms*. PhD thesis, ETH Zurich, 2000. URL: <https://cs.uwaterloo.ca/~astorjoh/diss2up.pdf>.
- 33 Nikoloz Tsimakuridze and Otfried Gühne. Graph states and local unitary transformations beyond local Clifford operations. *Journal of Physics A: Mathematical and Theoretical*, 50(19):195302, Apr 2017. [arXiv:1611.06938](#), doi:10.1088/1751-8121/aa67cd.
- 34 Ilan Tzitrin. Local equivalence of complete bipartite and repeater graph states. *Physical Review A*, 98(3):032305, 2018. [arXiv:1805.05968](#), doi:10.1103/PhysRevA.98.032305.
- 35 Bei Zeng, Hyeyoun Chung, Andrew W. Cross, and Isaac L. Chuang. Local unitary versus local Clifford equivalence of stabilizer and graph states. *Physical Review A*, 75(3), Mar 2007. [arXiv:quant-ph/0611214](#), doi:10.1103/physreva.75.032325.

A Alternative proof of Proposition 7

Recall that G and G' are said LC-equivalent if there exists a sequence of (possibly repeating) vertices a_1, \dots, a_m such that $G' = G \star a_1 \star \dots \star a_m$. According to Bouchet, G and G' are LC-equivalent if and only if there exist $A, B, C, D \subseteq V$ such that

- (i) $\forall u, v \in V$,
 $|B \cap N_G(u) \cap N_{G'}(v)| + |A \cap N_G(u) \cap \{v\}| + |D \cap \{u\} \cap N_{G'}(v)| + |C \cap \{u\} \cap \{v\}| = 0 \pmod 2$
- (ii) $A \cap D \Delta B \cap C = V$

We begin by proving by induction the "only if" part of the statement. First, notice that equations (i) and (ii) are satisfied when $G = G'$ with $A = D = V$ and $B = C = \emptyset$. Indeed, let $u, v \in V$:

$$\begin{aligned}
 & |B \cap N_G(u) \cap N_G(v)| + |A \cap N_G(u) \cap \{v\}| + |D \cap \{u\} \cap N_G(v)| + |C \cap \{u\} \cap \{v\}| \\
 &= |N_G(u) \cap \{v\}| + |\{u\} \cap N_G(v)| \\
 &= 0 \pmod 2
 \end{aligned}$$

Furthermore, $A \cap D \Delta B \cap C = V$.

Now, suppose that G and G' are equivalent and there exist $A, B, C, D \subseteq V$ satisfying (i) and (ii). Applying a local complementation on some vertex w in G results in the graph $G \star w$, which is also LC-equivalent to G' . Define

- $A' = A \Delta (\{w\} \cap C)$
- $B' = B \Delta (\{w\} \cap D)$
- $C' = C \Delta (N_G(w) \cap A)$
- $D' = D \Delta (N_G(w) \cap B)$

Let us show that $A', B', C', D' \subseteq V$ satisfy (i) and (ii) for the graphs $G \star w$ and G' .

Proof that A', B', C', D' satisfy (i).

Let $u, v \in V$.

$$\begin{aligned} & |B' \cap N_{G \star w}(u) \cap N_{G'}(v)| + |A' \cap N_{G \star w}(u) \cap \{v\}| + |D' \cap \{u\} \cap N_{G'}(v)| + |C' \cap \{u\} \cap \{v\}| \\ &= |(B\Delta(\{w\} \cap D)) \cap N_{G \star w}(u) \cap N_{G'}(v)| + |(A\Delta(\{w\} \cap C)) \cap N_{G \star w}(u) \cap \{v\}| \\ &\quad + |(D\Delta(N_G(w) \cap B)) \cap \{u\} \cap N_{G'}(v)| + |(C\Delta(N_G(w) \cap A)) \cap \{u\} \cap \{v\}| \end{aligned}$$

If $u \not\sim_G w$, then $N_{G \star w}(u) = N_G(u)$ and $N_G(w) \cup \{u\} = N_G(u) \cup \{w\} = \emptyset$:

$$\begin{aligned} &= |B \cap N_G(u) \cap N_{G'}(v)| + |A \cap N_G(u) \cap \{v\}| + |D \cap \{u\} \cap N_{G'}(v)| + |C \cap \{u\} \cap \{v\}| \\ &\quad + |\{w\} \cap D \cap N_G(u) \cap N_{G'}(v)| + |\{w\} \cap C \cap N_G(u) \cap \{v\}| \\ &\quad + |N_G(w) \cap B \cap \{u\} \cap N_{G'}(v)| + |N_G(w) \cap A \cap \{u\} \cap \{v\}| \bmod 2 \\ &= 0 \bmod 2 \end{aligned}$$

If $u \sim_G w$, then $N_{G \star w}(u) = N_G(u) \Delta N_G(w) \Delta \{u\}$, $N_G(w) \cup \{u\} = \{u\}$ and $N_G(u) \cup \{w\} = \{w\}$, thus :

$$\begin{aligned} &= |(B\Delta(\{w\} \cap D)) \cap (N_G(u) \Delta N_G(w) \Delta \{u\}) \cap N_{G'}(v)| \\ &\quad + |(A\Delta(\{w\} \cap C)) \cap (N_G(u) \Delta N_G(w) \Delta \{u\}) \cap \{v\}| \\ &\quad + |(D\Delta(N_G(w) \cap B)) \cap \{u\} \cap N_{G'}(v)| + |(C\Delta(N_G(w) \cap A)) \cap \{u\} \cap \{v\}| \\ &= |B \cap N_G(u) \cap N_{G'}(v)| + |A \cap N_G(u) \cap \{v\}| + |D \cap \{u\} \cap N_{G'}(v)| + |C \cap \{u\} \cap \{v\}| \\ &\quad + |B \cap N_G(w) \cap N_{G'}(v)| + |A \cap N_G(w) \cap \{v\}| + |D \cap \{w\} \cap N_{G'}(v)| + |C \cap \{w\} \cap \{v\}| \\ &\quad + |B \cap \{u\} \cap N_{G'}(v)| + |A \cap \{u\} \cap \{v\}| + |B \cap \{u\} \cap N_{G'}(v)| + |A \cap \{u\} \cap \{v\}| \\ &\quad + |\{w\} \cap D \cap N_G(w) \cap N_{G'}(v)| + |\{w\} \cap C \cap N_G(w) \cap \{v\}| \\ &\quad + |\{w\} \cap D \cap \{u\} \cap N_{G'}(v)| + |\{w\} \cap C \cap \{u\} \cap \{v\}| \\ &= 0 \bmod 2 \end{aligned}$$

Proof that A', B', C', D' satisfy (ii).

$$\begin{aligned} & (A' \cap D') \Delta (B' \cap C') \\ &= ((A\Delta(\{w\} \cap C)) \cap (D\Delta(N_G(w) \cap B))) \Delta ((B\Delta(\{w\} \cap D)) \cap (C\Delta(N_G(w) \cap A))) \\ &= (A \cap D) \Delta (A \cap N_G(w) \cap B) \Delta (D \cap \{w\} \cap C) \Delta (B \cap C) \Delta (B \cap N_G(w) \cap A) \Delta (C \cap \{w\} \cap D) \\ &= (A \cap D) \Delta (B \cap C) = V \end{aligned}$$

Now we prove the "if" part of the statement. The proof is very similar to a proof in [11] regarding the relation between local complementation and local Clifford operators. Let G and G' be two graphs defined on the same vertex set V along with A, B, C, D satisfying (i) and (ii). Condition (ii) implies that for some vertex $u \in V$, 6 cases can occur:

1. $u \in A \cap \overline{B} \cap \overline{C} \cap D$;
2. $u \in A \cap B \cap \overline{C} \cap D$;
3. $u \in A \cap \overline{B} \cap C \cap D$;
4. $u \in \overline{A} \cap B \cap C \cap \overline{D}$;
5. $u \in A \cap B \cap C \cap \overline{D}$;
6. $u \in \overline{A} \cap B \cap C \cap D$.

We call V_1 (resp. V_2, V_3, V_4, V_5, V_6) the set of vertices in case 1 (resp. 2, 3, 4, 5, 6). Notice that $V = V_1$ implies $G = G'$, indeed condition (ii) implies that for any $u, v \in V$, $|N_G(u) \cap \{v\}| + |\{u\} \cap N_{G'}(v)| = 0 \bmod 2$ i.e. $u \sim_G v \Leftrightarrow u \sim_{G'} v$. Furthermore, applying a local complementation on a vertex w of G changes the sets A, B, C, D , thus it changes in which case a vertex is. The changes are given in the following table (the case in which unwritten vertices are remain unchanged).

Case of w in		Case of $u \in N_G(w)$ in		Case of w_1 (or w_2) in	
G	$G \star w$	G	$G \star w$	G	$G \wedge w_1 w_2$
1	2	1	3	1	4
2	1	2	5	2	6
3	6	3	1	3	5
4	5	4	6	4	1
5	4	5	2	5	3
6	3	6	4	6	2

The table indicates that if G and G' are LC-equivalent and A, B, C, D satisfy (i) and (ii), then, for $G \wedge w_1 w_2$ and G', A', B', C', D' satisfy (i) and (ii) where:

- $A' = (A \cap \setminus \{w_1, w_2\}) \cup (C \cap \{w_1, w_2\})$
- $B' = (B \cap \setminus \{w_1, w_2\}) \cup (D \cap \{w_1, w_2\})$
- $C' = (C \cap \setminus \{w_1, w_2\}) \cup (A \cap \{w_1, w_2\})$
- $D' = (D \cap \setminus \{w_1, w_2\}) \cup (B \cap \{w_1, w_2\})$

Let us design an algorithm that produces a sequence of (possibly repeating) vertices $s = (a_1, \dots, a_m)$ such that $G' = G \star a_1 \star \dots \star a_m$. Initialise $G_0 = G$, $s_0 = []$ an empty sequence of vertices and $A_0 = A$, $B_0 = B$, $C_0 = C$, $D_0 = D$.

1. If there is a vertex u in case 2 or 6: let $s_0 \leftarrow s_0 + [u]$, $G_0 \leftarrow G_0 \star u$, $A_0 \leftarrow A_0 \Delta (\{w\} \cap C_0)$, $B_0 \leftarrow B_0 \Delta (\{w\} \cap D_0)$, $C_0 \leftarrow C_0 \Delta (N_{G_0}(w) \cap A_0)$, $D_0 \leftarrow D_0 \Delta (N_{G_0}(w) \cap B_0)$. Repeat until there is no vertex in case 2 or 6 left.
2. If there is a vertex u in case 4 or 5: let $v \in N_{G_0}(u)$ such that v is also in case 4 or 5. Let $s_0 \leftarrow s_0 + [u, v, u]$, $G_0 \leftarrow G_0 \wedge uv$, $A_0 \leftarrow (A_0 \cap \setminus \{u, v\}) \cup (C_0 \cap \{u, v\})$, $B_0 \leftarrow (B_0 \cap \setminus \{u, v\}) \cup (D_0 \cap \{u, v\})$, $C_0 \leftarrow (C_0 \cap \setminus \{u, v\}) \cup (A_0 \cap \{u, v\})$, $D_0 \leftarrow (D_0 \cap \setminus \{u, v\}) \cup (B_0 \cap \{u, v\})$. Then go to step 1.

Correctness. The evolution of A_0 , B_0 , C_0 and D_0 at each iteration of the algorithm ensures that (i) and (ii) are satisfied for $G \star s_0$ and G' . In step 2, if there is a vertex u in case 4 or 5, let us show that there exists $v \in N_{G_0}(u)$ such that v is also in case 4 or 5. Notice that in step 2, no vertex is in case 2 or 6. Suppose by contradiction that every vertex in $N_{G_0}(u)$ is in case 1 or 3, i.e. for every $v \in N_{G_0}(u)$, $v \in A \cap \bar{B} \cap D$. Then $|B_0 \cap N_{G_0}(u) \cap N_{G'}(u)| + |A_0 \cap N_{G_0}(u) \cap \{u\}| + |D_0 \cap \{u\} \cap N_{G'}(u)| + |C_0 \cap \{u\} \cap \{u\}| = |C_0 \cap \{u\}| = 1 \pmod 2$, contradicting (ii). At the end of the algorithm, every vertex is in case 1 or 3. Actually, every vertex is in case 1. Suppose by contradiction there is a vertex u in case 3. Then $|B_0 \cap N_{G_0}(u) \cap N_{G'}(u)| + |A_0 \cap N_{G_0}(u) \cap \{u\}| + |D_0 \cap \{u\} \cap N_{G'}(u)| + |C_0 \cap \{u\} \cap \{u\}| = |C_0 \cap \{u\}| = 1 \pmod 2$, contradicting (ii). Thus, at the end of the algorithm, $G' = G \star s_0$.

Termination. The number of vertices in case 1 or 3 strictly increases at each iteration of the algorithm.

B Interpretation of the constraints in terms of local Clifford operators

There is a one-to-one correspondence between the solutions to equations (i) and (ii) and the local Clifford operators (up to Pauli operators) that maps $|G\rangle$ to $|G'\rangle$. In particular if A, B, C, D satisfy equations (i) and (ii), then $|G'\rangle = e^{i\theta} \bigotimes_{v \in V} U_v |G\rangle$ where for any $v \in V$, U_v is equal, up to a Pauli operator, to:

More details on the LC-equivalence of graphs and the corresponding Clifford operators can be found in [13, 18]. From a graph state point of view, Proposition 10 provides an efficient algorithm to decide whether two graph states are LC-equivalent under some constraints on

I	if $v \in \overline{B} \cap \overline{C}$	H	if $v \in \overline{A} \cap \overline{D}$
$Z(\pi/2)$	if $v \in \overline{B} \cap C$	$Z(\pi/2)H$	if $v \in \overline{A} \cap D$
$X(\pi/2)$	if $v \in B \cap \overline{C}$	$X(\pi/2)H$	if $v \in A \cap \overline{D}$

■ **Table 1** Corresponding Clifford operators.

the Clifford operators. Notice that such constraints should be expressible as a linear equation through the correspondence given in Table 1. We give below a non-exhaustive family of constraints expressible as linear equations (in the following, k denotes an integer).

- $v \notin B$: U_v is $Z(k\pi/2)$ up to Pauli;
- $v \notin C$: U_v is $X(k\pi/2)$ up to Pauli;
- $v \notin A$: U_v is $Z(k\pi/2)H$ up to Pauli;
- $v \notin D$: U_v is $X(k\pi/2)H$ up to Pauli;
- $v \notin \overline{B} \cap \overline{C}$: U_v is a Pauli;
- $v \notin \overline{A} \cap \overline{D}$: U_v is H up to a Pauli;
- $v \in A$ iff $v \in D$: U_v is I , $X(\pi/2)$, $Z(\pi/2)$ or H up to Pauli, i.e. U_v^2 is a Pauli;
- $v \in A$ iff $v \in B$: U_v is $X(\pi/2)$ or $X(k\pi/2)H$ up to Pauli;
- $v \in A$ iff $w \in A$, $v \in B$ iff $w \in B$, $v \in C$ iff $w \in C$, $v \in D$ iff $w \in D$: $U_v = U_w$ up to Pauli.

C Proof of Lemma 16

► **Lemma 16.** *There exists an efficient algorithm that takes as inputs two graphs G_1 and G_2 of order n , and either claim that they are not LU -equivalent, or compute an MLS cover \mathcal{M} and two graphs G'_1 and G'_2 LC -equivalent to G_1 and G_2 respectively, such that G'_1 and G'_2 are both in standard form with respect to \mathcal{M} , in runtime $O(n^{6.38})$.*

Proof. TOPROVE 8 ◀

► **Remark 35.** In the main algorithm described in Section 3.3, the number of vertices of type Z is directly linked to the runtime of the algorithm. Thus, it is preferable that the number of vertices of type Z is low. It should be noted that at any point of the algorithm, following [11], if there exists a set K of vertices of type Z of size more than $\lfloor n/2 \rfloor + 1$, then a minimal local set L within K can be found, and adding L to \mathcal{M} transforms at least one vertex of type Z into a vertex of type \perp . Repeating the operation leads to a graph where the number of vertices of type Z is at most $\lfloor n/2 \rfloor$. The algorithm can then be restarted from step 2. Notice that the number of vertices of type \perp never decreases, thus this procedure will be performed less than n times. Adding random minimal local sets to \mathcal{M} is another way of trying to reduce the number of vertices of type Z .

D Proof of Lemma 20

► **Lemma 20.** *G_1 and G_2 are LC_r -equivalent if and only if there exists a sequence of (possibly repeating) vertices a_1, \dots, a_m such that $G_2^\# = G_1^\# \star a_1 \star \dots \star a_m$ satisfying the following additional constraints:*

- *the sequence contains no vertex of V_Z ;*
- *for each $\omega \in \mathcal{B}$, either the sequence contains every vertex of \mathcal{P}^ω exactly once, or it contains none.*

Proof. [TOPROVE 9](#)

E Proof of Lemma 26

► **Lemma 26.** *If $G \star^r S$ is valid and there is no S' such that $G \star^{r-1} S' = G \star^r S$, then $|V \setminus \text{supp}(S)| \geq r + 3$.*

Proof. [TOPROVE 10](#)

F Computer-assisted study of 2-local complementation

We use the following lemma to drastically decrease the size of the space to explore when studying 2-local complementation.

► **Lemma 36.** *Let S be a 2-incident independent multiset of a graph $G = (V, E)$ and suppose that there exists no set $A \subseteq \text{supp}(S)$ such that $G \star^2 S = G \star^1 A$. Then there exists a graph $G' = (V', E')$ bipartite with respect to a bipartition $S', V' \setminus S'$ of the vertices such that:*

- S' is 2-incident;
- S' contains no twins;
- S' contains no vertex of degree 0 or 1;
- $|S'| \leq |\text{supp}(S)|$;
- $|V' \setminus S'| \leq |V \setminus \text{supp}(S)|$;
- there exists no set $A \subseteq S'$ such that $G' \star^2 S' = G' \star^1 A$.

Proof. [TOPROVE 11](#)

Given an integer k , let \mathcal{G}_k be the class of graphs that are bipartite with respect to a bipartition $S, V \setminus S$ of the vertices such that:

- S is 2-incident;
- S contains no twins;
- S contains no vertex of degree 0 or 1;
- $|V \setminus S| = k$.

It is easy to generate each graph of \mathcal{G}_k , although the number of elements in \mathcal{G}_k grows double exponentially fast with k . S can be defined as a list of words in $\{0, 1\}^k$ of weight at least 2. More precisely each vertex of S is uniquely associated with a set of $V \setminus S$ of size at least 2, its neighbourhood. Furthermore, the 2-incidence of S implies that S is uniquely determined by the set of its vertices of degree at least 4. Indeed, starting from a set containing only vertices of degree at least 4, the conditions of the form " $S \bullet \Lambda_G^K = 0 \bmod 2^{r-k-\delta(k)}$ " translate into a procedure to find which vertices of degree 3 then 2 need to be added to the set so that S is 2-incident. This proves that there is a bijection between \mathcal{G}_k and lists of words in $\{0, 1\}^k$ of weight at least 4. Thus, the size of \mathcal{G}_k is exactly given by the formula

$$|\mathcal{G}_k| = 2^{\binom{k}{4} + \binom{k}{5} + \dots + \binom{k}{k}}$$

For $k = 1, 2, 3, 4, 5$ and 6 , the size of \mathcal{G}_k is respectively $1, 1, 1, 2, 2^6 = 64$, and $2^{22} \sim 4 \times 10^6$ which is suitable for computation. But, even for k as low as seven, the size of \mathcal{G}_7 is $2^{64} \sim 2 \times 10^{19}$.

For every k from 1 to 6, we generate each graph G of \mathcal{G}_k , along with the set S defined above. Notice that a local complementation over a vertex u of S toggles the connectivity of some pairs of vertices of $V \setminus S$, here the pairs where each end is a neighbour of u . In other

words, to each vertex u of S we associate a vector in $\mathbb{F}_2^{\binom{k}{2}}$ corresponding to the action of the local complementation of u on the graph. The set of the vectors corresponding to each vertex of S spans a \mathbb{F}_2 -vector space \mathcal{L} describing the action of local complementation over vertices of S on the graph. Using Gaussian elimination, we are able to compute a basis of \mathcal{L} . Furthermore, we compute the vector x in $\mathbb{F}_2^{\binom{k}{2}}$ corresponding to the action of a 2-local complementation over S on the graph. Checking if the action of a 2-local complementation over S can be implemented by local complementations on vertices of S amounts to checking if x belongs to the vector space \mathcal{L} , which can be done efficiently using Gaussian elimination.

For $k \in [1, 3]$, the set S corresponding to the only graph $G \in \mathcal{G}_k$ is empty, hence a 2-local complementation over S leaves G invariant, i.e. $G \star^2 S = G$. For $k = 4$, S is either empty or contains 11 vertices; in both case it is easy to check that $G \star^2 S = G$. For $k = 5$, the computation shows that for each graph $G \in \mathcal{G}_5$, $G \star^2 S = G$. Now, fix $k = 6$. For each graph $G \in \mathcal{G}_6$ such that the corresponding set S contains at most 16 vertices, $G \star^2 S = G$. For each graph $G \in \mathcal{G}_6$ such that the corresponding set S contains at most 20 vertices, a 2-local complementation over the corresponding set S can be implemented by local complementations over vertices of S , i.e. there exists a set $A \subseteq S$ such that $G \star^2 S = G \star^1 A$. The property does not hold if S is of size 21, for instance we recover the well-known 27-vertex counterexample to the LU=LC conjecture described in [33].

According to Lemma 36, this is enough to prove Lemma 31:

► **Lemma 31.** *Let S be a 2-incident independent multiset of a graph G . If $|V \setminus \text{supp}(S)| \leq 5$, or if $|V \setminus \text{supp}(S)| = 6$ and $|\text{supp}(S)| \leq 20$, then a 2-local complementation on S can be implemented by local complementations over a subset of $\text{supp}(S)$.*

G Proof of Lemma 32

► **Lemma 32.** *Let S be a 2-incident independent set of a graph G such that S does not contain any twins and $|S| \leq 12$. Then, $G \star^2 S = G$.*

Proof. TOPROVE 12

◀