Partial complementation of graphs

Fedor V. Fomin, Petr A. Golovach, Torstein Strømme, Dimitrios M. Thilikos

Partial complementation of graphs*

Fedor V. Fomin[1], Petr A. Golovach[1], Torstein J. F. Strømme[1], and Dimitrios M. Thilikos [2,3]

1 **Department of Informatics, University of Bergen, Norway**
{fedor.fomin, petr.golovach, torstein.stromme}@ii.uib.no
2 **AlGCo project, CNRS, LIRMM, France,** sedthilk@thilikos.info
3 **Department of Mathematics National and Kapodistrian University of Athens, Greece**

Abstract

A *partial complement* of the graph G is a graph obtained from graph G by complementing edges of some of its induced subgraphs. We study the following algorithmic question: for a given graph G and graph class $\mathcal{G}$, is it possible to partially complement G to $\mathcal{G}$? We show that this problem can be solved in polynomial time for various classes of graphs like bipartite, degenerate, or cographs. We complement these results by proving that the problem is NP-complete when $\mathcal{G}$ is the class of r-regular graphs.

Digital Object Identifier 10.4230/LIPIcs...

1 Introduction

One of the most important questions in Graph Theory concerns the efficiency of recognition of a graph class $\mathcal{G}$. For example, how fast we can decide whether a graph is chordal, 2-connected, triangle-free, of bounded treewidth, bipartite, 3-colorable, or excludes some fixed graph as a minor? In particular, the recent developments in parameterized algorithms are driven by the problems of recognizing of graph classes which do not differ up to small "small disturbance" from graph classes recognizable in polynomial time. The amount of disturbance is quantified in "atomic" operations required for modifying an input graph into the "well-behaving" graph class $\mathcal{G}$. The standard operations could be edge/vertex deletions, additions or edge contractions. Many problems in graph algorithms fall into this graph modification category: is it possible to add at most k edges to make a graph 2-edge connected or to make it chordal? Or is it possible to delete at most k vertices such that the resulting graph has no edges or contains no cycles?

A rich subclass of modification problems concerns edge editing problems. Here the "atomic" operation is the change of adjacency, i.e. for a pair of vertices u, v, we can either add an edge uv or delete edge uv. For example, the Cluster Editing problem asks to transform an input graph into a cluster graph, that is a disjoint union of cliques, by at most k adjacency relations.

The *complement* of a graph G is a graph H on the same vertices such that two distinct vertices of H are adjacent if and only if they are not adjacent in G. In this paper we study the partial complement of a graph, which was introduced by Kamiński, Lozin, and Milanič in [9] in their study of the clique-width of a graph.

The *partial complement* of a graph is a graph obtained from graph G by complementing edges of some of its induced subgraphs. More formally, for a graph G and $S \subseteq V(G)$, we

* The first three authors have been supported by the Research Council of Norway via the projects "CLASSIS" and "MULTIVAL". The fourth author has been supported by project "DEMOGRAPH" (ANR-16-CE40-0028).

define $G \oplus S$ as the graph with the vertex set $V(G)$ whose edge set is defined as follows: a pair of distinct vertices u, v is an edge of $G \oplus S$ if and only if one of the following holds:

- either $uv \in E(G)$ and at least one vertex from $\{u, v\}$ does not belong to S, or
- $u, v \in S$ and $uv \notin E(G)$.

Thus when set S consists only of two vertices $\{u, v\}$, then the operation changes the adjacency between u and v, and for larger set S, $G \oplus S$ changes the adjacency relations for all pairs of vertices of S.

Finally, we say that a graph H is a *partial complement of the graph* G if H is isomorphic to $G \oplus S$ for some $S \subseteq V(G)$. For a graph class $\mathcal{G}$ and graph G, we say that there is a *partial complement of* G *to* $\mathcal{G}$ if for some $S \subseteq V(G)$, we have $G \oplus S \in \mathcal{G}$.

Let $\mathcal{G}$ be a graph class. We consider the following generic algorithmic problem.

Partial Complement to $\mathcal{G}$ (PC$\mathcal{G}$)
Input: A simple undirected graph G.
Question: Is there a partial complement of G to $\mathcal{G}$?

In other words, how difficult is it to recognize the class $\mathcal{G} \oplus S$, which is the class of graphs such that each graph in this class can be partially complemented into $\mathcal{G}$? In this paper we show that there are many well-known graph classes $\mathcal{G}$ such that $\mathcal{G} \oplus S$ are recognizable in polynomial time. We show that

- Partial Complement to $\mathcal{G}$ is solvable in time $\mathcal{O}(f(n) \cdot n^4 + n^6)$ when $\mathcal{G}$ is a triangle-free graph class recognizable in time $f(n)$. For example, this implies that when $\mathcal{G}$ is the class of bipartite graphs, the class $\mathcal{G} \oplus S$ is recognizable in polynomial time.
- Partial Complement to $\mathcal{G}$ is solvable in time $f(n) \cdot n^{\mathcal{O}(1)}$ when $\mathcal{G}$ is a d-degenerate graph class recognizable in time $f(n)$. Thus when $\mathcal{G}$ is the class of planar graphs, class of cubic graphs, class of graph of bounded treewidth, or class of H-minor free graphs, then the class $\mathcal{G} \oplus S$ is recognizable in polynomial time.
- Partial Complement to $\mathcal{G}$ is solvable in polynomial time when $\mathcal{G}$ is a class of bounded clique-width expressible in monadic second-order logic (with no edge set quantification). Therefore, if $\mathcal{G}$ is the class of P_4-free graphs, class $\mathcal{G} \oplus S$ is recognizable in polynomial time.
- Partial Complement to $\mathcal{G}$ is solvable in polynomial time when $\mathcal{G}$ can be described by a 2×2 M-partition matrix. This implies in particular, that $\mathcal{G} \oplus S$ is recognizable in polynomial time, where $\mathcal{G}$ is the class of split graphs.

Nevertheless, there are cases when the problem is NP-hard. In particular, we prove that this holds when $\mathcal{G}$ is the class of r-regular graphs.

2 Partial complementation to triangle-free graph classes

A triangle is a complete graph on three vertices. Many graph classes does not allow the triangle as a subgraph, for instance trees, forests, or graphs with large girth. In this paper we show that partial complementation to triangle-free graphs can be decided in polynomial time.

More precisely, we show that if a graph class $\mathcal{G}$ can be recognized in polynomial time and it is triangle-free, then we can also solve Partial Complement to $\mathcal{G}$ in polynomial time.

Our algorithm is constructive, and returns a *solution* $S \subseteq V(G)$, that is a set S such that $G \oplus S$ is in $\mathcal{G}$. We say that a solution *hits* an edge uv (or a non-edge $\overline{uv}$), if both u and v are contained in S.

Our algorithm considers each of the following cases.

(i) There is a solution S of size at most two.
(ii) There is a solution S containing two vertices that are non-adjacent in G.
(iii) There is a solution S such that it form a clique of size at least 3 in G.
(iv) G is a no-instance.

Case (i) can be resolved in polynomial time by brute-force, and thus we start from analyzing the structure of a solution in Case (ii). We need the following observation.

▶ Observation 1. Let $\mathcal{G}$ be a class of triangle-free graphs and let G be an instance of Partial Complement to $\mathcal{G}$, where $S \subseteq V(G)$ is a valid solution. Then

a) $G[S]$ does not contain an independent set of size 3, and
b) for every triangle $\{u, v, w\} \subseteq V(G)$, at least two vertices are in S.

Because all non-edges between vertices in $G[S]$ become edges in $G \oplus S$ and vice versa, whereas all (non-) edges with an endpoint outside S remain untouched, we see that the observation holds.

Let us recall that a graph G is a *split graph* if its vertex set can partitioned into $V(G) = C \cup I$, where C is a clique and I is an independent set. Let us note that the vertex set of a split graph can have several *split partitions*, i.e. partitions into a clique and independent set. However, the number of split partitions of an n-vertex split graphs is at most n. The analysis of Case (ii) is based on the following lemma.

▶ **Lemma 1.** *Let $\mathcal{G}$ be a class of triangle-free graphs and let G be an instance of* Partial Complement to $\mathcal{G}$. *Let $S \subseteq V(G)$ be a valid solution which is not a clique, and let $u, v \in S$ be distinct vertices such that $uv \notin E(G)$. Then*

a) the entire solution S is a subset of the union of the closed neighborhoods of u and v, that is $S \subseteq N_G[u] \cup N_G[v]$;
b) every common neighbor of u and v must be contained in the solution S, that is $N_G(u) \cap N_G(v) \subseteq S$;
c) the graph $G[N(u) \setminus N(v)]$ is a split graph. Moreover, $(N(u) \setminus N(v)) \cap S$ is a clique and $(N(u) \setminus N(v)) \setminus S$ is an independent set.

Proof. We will prove each point separately, and in order.

a) Assume for the sake of contradiction that the solution S contains a vertex $w \notin N_G[u] \cup N_G[v]$. But then $\{u, v, w\}$ is an independent set in G, which contradicts item a) of Observation 1.
b) Assume for the sake of contradiction that the solution S does not contain a vertex $w \in N_G(u) \cap N_G(v)$. Then the edges uw and vw will both be present in $G \oplus S$, as well as the edge uv. Together, these form a triangle.
c) We first claim that the solution S is a vertex cover for $G[N(u) \setminus N(v)]$. If it was not, then there would exist an edge u_1u_2 of $G[N(u) \setminus N(v)]$ such that both endpoints $u_1, u_2 \notin S$, yet u_1, u_2 would form a triangle with u in $G \oplus S$, which would be a contradiction. Hence $(N(u) \setminus N(v)) \setminus S$ is an independent set. Secondly, we claim that $(N(u) \setminus N(v)) \cap S$ forms a clique. If not, then there would exist $u_1, u_2 \in (N(u) \setminus N(v)) \cap S$ which are nonadjacent. In this case $\{u_1, u_2, v\}$ is an independent set, which contradicts item a) of Observation 1. Taken together, these claims imply the last item of the lemma.

◀

We now move on to examine the structure of a solution for the third case, when there exists a solution which is a clique of size at least three.

▶ **Lemma 2.** *Let $\mathcal{G}$ be a class of triangle-free graphs and let G be an instance of* Partial Complement to $\mathcal{G}$. *Let $S \subseteq V(G)$ be a solution such that $|S| \geq 3$ and $G[S]$ is a clique. Let $u, v \in S$ be distinct. Then*

a) the solution S is contained in their common neighborhood, that is $S \subseteq N_G[u] \cap N_G[v]$, and
b) the graph $G[N_G[u] \cap N_G[v]]$ is a split graph where $(N_G[u] \cap N_G[v]) \setminus S$ is an independent set.

Proof. We prove each point separately, and in order.

a) Assume for the sake of contradiction that the solution S contains a vertex w which is not in the neighborhood of both u and v. This contradicts that S is a clique.
b) We claim that S is a vertex cover of $G[N_G[u] \cap N_G[v]]$. Because S is also a clique, the statement of the lemma will then follow immediately. Assume for the sake of contradiction that S is not a vertex cover. Then there exist an uncovered edge w_1w_2, where $w_1, w_2 \in N_G[u] \cap N_G[v]$, and also $w_1, w_2 \notin S$. Since $\{u, w_1, w_2\}$ form a triangle, we have by b) of Observation 1 that at least two of these vertices are in S. That is a contradiction, so our claim holds.

◀

We now have everything in place to present the algorithm.

▶ Algorithm 1 (Partial Complement to $\mathcal{G}$ where $\mathcal{G}$ is triangle-free).
Input: An instance G of PC$\mathcal{G}$ where $\mathcal{G}$ is a triangle-free graph class recognizable in time $f(n)$ for some function f.
Output: A set $S \subseteq V(G)$ such that $G \oplus S$ is in $\mathcal{G}$, or a correct report that no such set exists.

1. By brute force, check if there is a solution of size at most 2. If yes, return this solution.
2. For every non-edge $\overline{uv}$ of G:
   a. If either $G[N(u) \setminus N_G(v)]$ or $G[N_G(u) \setminus N_G(v)]$ is not a split graph, skip this iteration and try the next non-edge.
   b. Let (I_u, C_u) and (I_v, C_v) denote a split partition of $G[N_G(u) \setminus N_G(v)]$ and $G[N_G(v) \setminus N_G(u)]$ respectively. For each pair of split partitions $(I_u, C_u), (I_v, C_v)$:
      i. Construct solution candidate $S' := \{u, v\} \cup (N_G(u) \cap N_G(v)) \cup C_u \cup C_v$
      ii. If $G \oplus S'$ is a member of $\mathcal{G}$, return S'
3. Find a triangle $\{x, y, z\}$ of G
4. For each edge in the triangle $uv \in \{xy, xz, yz\}$:
   a. If $G[N_G(u) \cap N_G(v)]$ is not a split graph, skip this iteration and try the next edge.
   b. For each possible split partition (I, C) of $G[N_G(u) \cap N_G(v)]$:
      i. Construct solution candidate $S' := \{u, v\} \cup C$
      ii. If $G \oplus S'$ is a member of $\mathcal{G}$, return S'
5. Return 'None'

▶ **Theorem 3.** *Let $\mathcal{G}$ be a class of triangle-free graphs such that deciding whether an n-vertex graph is in $\mathcal{G}$ is recognizable in time $f(n)$ for some function f. Then* Partial Complement to $\mathcal{G}$ *is solvable in time $\mathcal{O}(n^6 + n^4 \cdot f(n))$.*

Proof. We will prove that Algorithm 1 is correct, and that its running time is $\mathcal{O}(n^4 \cdot (n^2 + f(n)))$. We begin by proving correctness. Step 1 is trivially correct. After Step 1 we can assume that any valid solution has size at least three, and we have handled Case (i) when there exists a solution of size at most two. We have the three cases left to consider: (ii)

There exists a solution which hits a non-edge, (iii) there is a solution S such that in $G \oplus S$ vertices of S form a clique of size at least 3, and (iv) no solution exists.

In the case that there exists a solution S hitting a non-edge uv, we will at some point guess this non-edge in Step 2 of the algorithm. By Lemma 1, we have that both $G[N_G(u) \setminus N_G(v)]$ and $G[N_G(u) \setminus N_G(v)]$ are split graphs, so we do not miss the solution S in Step 2a. Since we try every possible combinations of split partitions in Step 2b, we will by Lemma 1 at some point construct S' correctly such that $S' = S$.

In the case that there exist only solutions which hits exactly a clique, we first find some triangle $\{x, y, z\}$ of G. It must exist, since a solution S is a clique of size at least three. By Observation 1b, at least two vertices of the triangle must be in the S. At some point in step 4 we guess these vertices correctly. By Lemma 2b we know that $G[N_G(u) \cap N_G(v)]$ is a split graph, so we will not miss S in Step 4a. Since we try every split partition in Step 4b, we will by Lemma 2 at some point construct S' correctly such that $S' = S$.

Lastly, in the case that there is no solution, we know that there neither exists a solution of size at most two, nor a solution which hits a non-edge, nor a solution which hits a clique of size at least three. Since these three cases exhaust the possibilities, we can correctly report that there is no solution when none was found in the previous steps.

For the runtime, we start by observing that Step 1 takes time $\mathcal{O}(n^2 \cdot f(n))$. The sub-procedure of Step 2 is performed $\mathcal{O}(n^2)$ times, where step 2a takes time $\mathcal{O}(n \log n)$. The sub-procedure of Step 2b takes time at most $\mathcal{O}(n^2 + f(n))$, and it is performed at most $\mathcal{O}(n^2)$ times. In total, Step 2 will use no longer than $\mathcal{O}(n^4 \cdot (n^2 + f(n)))$ time. Step 3 is trivially done in time $\mathcal{O}(n^3)$. The sub-procedure of Step 4 is performed at most three times. Step 4a is done in $\mathcal{O}(n \log n)$ time, and step 4b is done in $\mathcal{O}(n \cdot (n^2 + f(n))$ time, which also becomes the asymptotic runtime of the entire step 4. The worst running time among these steps is Step 2, and as such the runtime of Algorithm 1 is $\mathcal{O}(n^4 \cdot (n^2 + f(n)))$. ◀

3 Complement to degenerate graphs

For $d > 0$, we say that a graph G is d-degenerate, if every induced (not necessarily proper) subgraph of G has a vertex of degree at most d. For example, trees are 1-degenerate, while planar graphs are 5-degenerate.

▶ **Theorem 4.** *Let $\mathcal{G}$ be a class of d-degenerate graphs such that deciding whether an n-vertex graph is in $\mathcal{G}$ is recognizable in time $f(n)$ for some function f. Then* Partial Complement to $\mathcal{G}$ *is solvable in time $f(n) \cdot n^{2^{\mathcal{O}(d)}}$.*

Proof. Let G be an n-vertex graph. We are seeking for a vertex subset S of G such that $G \oplus S \in \mathcal{G}$.

We start from trying all vertex subsets of G of size at most $2d$ as a candidate for S. Thus, in time $\mathcal{O}(n^{2d} \cdot f(n))$ we either find a solution or conclude that a solution, if it exists, should be of size more than $2d$.

Now we assume that $|S| > 2d$. We try all subsets of $V(G)$ of size $2d + 1$. Then if G can be complemented to $\mathcal{G}$, at least one of these sets, say X, is a subset of S. In total, we enumerate $\binom{n}{2d+1}$ sets.

First we consider the set Y of all vertices in $V(G) \setminus X$ with at least $d+1$ neighbors in X. The observation here is that most vertices from Y are in S. More precisely, if more than

$$\alpha = \binom{|X|}{d+1} \cdot d + 1 = \binom{2d+1}{d+1} \cdot d + 1$$

vertices of Y are not in S, then $G \oplus S$ contains a complete bipartite graph $G_{d+1,d+1}$ as a subgraph, and hence $G \oplus S$ is not d-degenerate. Thus, we make at most $\binom{n}{\alpha}$ guesses on which subset of Y is in S.

Similarly, when we consider the set Z of all vertices from $V(G) \setminus X$ with at most d neighbors in X, we have that at most α of vertices from Z could belong to S. Since $V(G) = X \cup Y \cup Z$, if there is a solution S, it will be found in at least one from

$$\binom{n}{2d+1} \cdot \alpha^2 = n^{2^{\mathcal{O}(d)}}$$

of the guesses. Since for each set S we can check in time $f(n)$ whether $G \oplus S \in \mathcal{G}$, this concludes the proof. ◀

4 Complement to M-partition

Many graph classes can be defined by whether it is possible to partition the vertices of graphs in the class such that certain internal and external edge requirements of the parts are met. For instance, a complete bipartite graph is one which can be partitioned into two sets such that every edge between the two sets is present (external requirement), and no edge exists within any of the partitions (internal requirements). Other examples are split graphs and k-colorable graphs. Feder et al. [6] formalized such partition properties of graph classes by making use of a symmetric matrix over $\{0, 1, \star\}$, called an *M-partition*.

▶ **Definition 5** (M-partition). For a $k \times k$ matrix M, we say that a graph G belongs to the graph class $\mathcal{G}_M$ if its vertices can be partitioned into k (possibly empty) sets $X_1, X_2, \ldots, X_k$ such that, for every $i \in [k]$, if

- $M[i,i] = 1$, then X_i is a clique and if $M[i,i] = 0$, then X_i is an independent set, and for every $i, j \in [k]$, $i \neq j$,
- if $M[i,j] = 1$, then every vertex of X_i is adjacent to all vertices of X_j,
- if $M[i,j] = 0$, then there is no edges between X_i and X_j.

Note that if $M[i,j] = \star$, then there is no restriction on the edges between vertices from X_i and X_j.

For example, for matrix

$$M = \begin{pmatrix} 0 & \star \\ \star & 0 \end{pmatrix}$$

the corresponding class of graphs is the class of bipartite graphs, while matrix

$$M = \begin{pmatrix} 0 & \star \\ \star & 1 \end{pmatrix}$$

identifies the class of split graphs.

In this section we prove the following theorem.

▶ **Theorem 6.** *Let $\mathcal{G} = \mathcal{G}_M$ be a graph class described by an M-partition matrix of size 2×2. Then* Partial Complement to $\mathcal{G}$ *is solvable in polynomial time.*

In particular, Theorem 6 yields polynomial algorithms for Partial Complement to $\mathcal{G}$ when $\mathcal{G}$ is the class of split graphs or (complete) bipartite graphs. The proof of our theorem is based on the following beautiful dichotomy result of Feder et al. [6] on the recognition of classes $\mathcal{G}_M$ described by 4×4 matrices.

▶ Proposition 1 ([6, Corollary 6.3]). Suppose M is a symmetric matrix over $\{0, 1, \star\}$ of size $k = 4$. Then the recognition problem for $\mathcal{G}_M$ is

- NP-complete when M contains the matrix for 3-coloring or its complement, and no diagonal entry is $\star$.
- Polynomial time solvable otherwise.

▶ **Lemma 7.** *Let M be a symmetric $k \times k$ matrix giving rise to the graph class $\mathcal{G}_M = \mathcal{G}$. Then there exists a $2k \times 2k$ matrix M' such that for any input G to* Partial Complement to $\mathcal{G}$*, it is a yes-instance if an only if G belongs to $\mathcal{G}_{M'}$.*

Proof. Given M, we construct a matrix M' in linear time. We let M' be a matrix of dimension $2k \times 2k$, where entry $M'[i, j]$ is defined as $M[\lceil \frac{i}{2} \rceil, \lceil \frac{j}{2} \rceil]$ if at least one of i, j is even, and $\neg M[\frac{i+1}{2}, \frac{j+1}{2}]$ if i, j are both odd. Here, $\neg 1 = 0$, $\neg 0 = 1$, and $\neg \star = \star$. For example, for matrix

$$M = \begin{pmatrix} 0 & \star \\ \star & 1 \end{pmatrix}$$

the above construction results in

$$M' = \begin{pmatrix} 1 & 0 & \star & \star \\ 0 & 0 & \star & \star \\ \star & \star & 0 & 1 \\ \star & \star & 1 & 1 \end{pmatrix}.$$

We prove the two directions separately.

($\Longrightarrow$) Assume there is a partial complementation $G \oplus S$ into $\mathcal{G}_M$. Let $X_1, X_2, \ldots, X_k$ be an M-partition of $G \oplus S$. We define partition $X'_1, X'_2, \ldots, X'_{2k}$ of G as follows. For every vertex $v \in X_i$, $1 \leq i \leq k$, we assign v to X'_{2i-1} if $v \in S$ and to X'_{2i} otherwise.

We now show that every edge of G respects the requirements of M'. Let $uv \in E(G)$ be an edge, and let $u \in X_i$ and $v \in X_j$. If at least one vertex from $\{u, v\}$, say v is not in S, then uv is also an edge in $G \oplus S$, thus $M[i, j] \neq 0$. Since $v \notin S$, it belongs to set $v \in X'_{2j}$. Vertex u is assigned to set X'_ℓ, where ℓ is either $2i$ or $2i - 1$, depending whether u belongs to S or not. But because $2j$ is even irrespectively of ℓ, $M'[\ell, 2j] = M[i, j] \neq 0$.

Now consider the case when both $u, v \in S$. Then the edge does not persist after the partial complementation by S, and thus $M[i, j] \neq 1$. We further know that u is assigned to X'_{2i-1} and v to X'_{2j-1}. Both $2i - 1$ and $2j - 1$ are odd, and by the construction of M', we have that $M'[2i - 1, 2j - 1] \neq 0$, and again the edge uv respects M'. An analogous argument shows that also all non-edges respect M'.

($\Longleftarrow$) Assume that there is a partition $X'_1, X'_2, \ldots, X'_{2k}$ of G according to M'. Let the set S consist of all vertices in odd-indexed parts of the partition. We now show that $G \oplus S$ can be partitioned according to M. We define partition $X_1, X_2, \ldots, X_k$ by assigning each vertex $u \in X'_i$ to $X_{\lceil \frac{i}{2} \rceil}$. It remains to show that $X_1, X_2, \ldots, X_k$ is an M-partition of $G \oplus S$.

Let $u \in X_i$, $v \in X_j$. Suppose first that $uv \in E(G \oplus S)$. If at least one of u, v is not in S, we assume without loss of generality that $v \notin S$. Then $uv \in E(G)$ and $v \in X'_{2j}$. For vertex $u \in X'_\ell$, irrespectively, whether ℓ is $2i$ or $2i - 1$, we have that $M'[\ell, 2j] = M[i, j] \neq 0$. But then $M[i, j] \neq 0$. Otherwise we have $u, v \in S$. Then uv is a non-edge in G, and thus $M'[2i - 1, 2j - 1] \neq 1$. But by the construction of M', we have that $M[i, j] \neq 0$, and there is no violation of M. An analogous argument shows that if u and v are not adjacent in $G \oplus S$, it holds that $M[i, j] \neq 1$. Thus $X_1, X_2, \ldots, X_k$ is an M-partition of $G \oplus S$, which concludes the proof. ◀

Now we are ready to prove Theorem 6.

Proof of Theorem 6. For a given matrix M, we use Lemma 7 to construct a matrix M'. Let us note that by the construction of matrix M', for every 2×2 matrix M we have that matrix M' has at most two 1's and at most two 0's along the diagonal. Then by Proposition 1, the recognition of whether G admits M'-partition is in P. Thus by Lemma 7, Partial Complement to $\mathcal{G}$ is solvable in polynomial time ◀

5 Partial complementation to graph classes of bounded clique-width

We show that Partial Complement to $\mathcal{G}$ can be solved in polynomial time when $\mathcal{G}$ has bounded clique-width and can be expressed by an $\mathbf{MSO}_1$ property. Definitions of clique-width, k-expressions and $\mathbf{MSO}_1$ are found in the appendix. We will use the following result of Hliněný and Oum [8].

▶ Proposition 2 ([8]). There is an algorithm that for every integer k and graph G in time $O(|V(G)|^3)$ either computes a $(2^{k+1}-1)$ expression for a graph G or correctly concludes that the clique-width of G is more than k.

Note that the algorithm of Hliněný and Oum only approximates the clique-width but does not provide an algorithm to construct an optimal k-expression tree for a graph G of clique-width at most k. But this approximation is usually sufficient for algorithmic purposes.

Courcelle, Makowsky and Rotics [2] proved that every graph property that can be expressed in $\mathbf{MSO}_1$ can be recognized in linear time for graphs of bounded clique-width when given a k-expression.

▶ Proposition 3 ([2, Theorem 4]). Let $\mathcal{G}$ be a class of graphs of clique-width at most k such that for each graph $G \in \mathcal{G}$, a corresponding k-expression can be found in $\mathcal{O}(f(n,m))$ time. Then every $\mathbf{MSO}_1$ property on $\mathcal{G}$ can be recognized in time $\mathcal{O}(f(n,m))$.

The nice property of graphs with bounded clique-width is that their partial complementation is also bounded. In particular, Kamiński, Lozin, and Milanič in [9] observed that if G is a graph of clique-width k, then any partial complementation of G is of clique-width at most $g(k)$ for some computable function g. For completeness, we provide a more accurate upper bound and sketch the proof (in the appendix).

▶ **Lemma 8** (∗). *Let G be a graph, $S \subseteq V(G)$. Then* $\textsc{cwd}(G \oplus S) \leq 3\textsc{cwd}(G)$.

For a class of graphs $\mathcal{G}$ we denote by $\mathcal{G}^{(1)}$ the class of graphs obtained from graphs in $\mathcal{G}$ by applying a partial complementation.

▶ **Lemma 9.** *Let φ be an $\mathbf{MSO}_1$ property describing the graph class $\mathcal{G}$. Then there exists an $\mathbf{MSO}_1$ property ϕ describing the graph class $\mathcal{G}^{(1)}$ of size $|\phi| \in \mathcal{O}(|\varphi|)$.*

Proof. We will construct ϕ from φ in the following way: We start by prepending $\exists S \subseteq V(G)$. Then for each assessment of the existence of an edge in φ, say $uv \in E(G)$, replace that term with $((u \notin S \vee v \notin S) \wedge uv \in E(G)) \vee (u \in S \wedge v \in S \wedge uv \notin E(G))$. Symmetrically, for each assessment of of the non-existence of an edge $uv \notin E(G)$, replace that term with $((u \notin S \vee v \notin S) \wedge uv \notin E(G)) \vee (u \in S \wedge v \in S \wedge uv \in E(G))$.

We observe that if φ is satisfiable for some graph G, then for every $S \subseteq V(G)$, the partial complementation $G \oplus S$ will yield a satisfying assignment to ϕ. Conversely, if ϕ is satisfiable for a graph G, then there exist some S such that φ is satisfied for $G \oplus S$. For the size, we note that each existence check for edges blows up by a constant factor. ◀

We are ready to prove the main result of this section.

▶ **Theorem 10.** *Let $\mathcal{G}$ be a graph class of bounded clique-width which can be expressed in MSO_1. Then* Partial Complement to $\mathcal{G}$ *is solvable in polynomial time.*

Proof. Let $\mathcal{G}$ be a graph class of clique-width at most k which can be expressed by an $\mathbf{MSO}_1$ formula φ. Let G be an n-vertex input graph. We apply Proposition 2 for G and in time $O(n^3)$ either obtain a $(2^{3k+1}-1)$ expression for G or conclude that the clique-width of G is more than $3k$. In the latter case, by Lemma 8, G cannot be partially complemented to $\mathcal{G}$.

We then obtain an $\mathbf{MSO}_1$ formula ϕ from Lemma 9, and apply Proposition 3, which works in time $f(k,\phi)\cdot n$ for some function f. In total, the runtime of the algorithm is $f(k,\phi)\cdot n + n^3$. ◀

We remark that if clique-width expression is provided along with the input graphs, and $\mathcal{G}$ can be expressed in $\mathbf{MSO}_1$, then there is a linear time algorithm for Partial Complement to $\mathcal{G}$. This follows directly from Lemma 9 and Proposition 3.

Theorem 10 implies that for every class of graphs $\mathcal{G}$ of bounded clique-width characterized by a finite set of finite forbidden induced subgraphs, e. g. P_4-free graphs (also known as cographs) or classes of graphs discussed in [1], the Partial Complement to $\mathcal{G}$ problem is solvable in polynomial time. However, Theorem 10 does not imply that Partial Complement to $\mathcal{G}$ is solvable in polynomial time for $\mathcal{G}$ being of clique-width at most k. This is because such a class $\mathcal{G}$ cannot be described by $\mathbf{MSO}_1$. Interestingly, for the related class $\mathcal{G}$ of graphs of bounded *rank-width* (see [3] for the definition) at most k, the result of Oum and Courcelle [4] combined with Theorem 10 implies that Partial Complement to $\mathcal{G}$ is solvable in polynomial time.

6 Hardness of partial complementation to r-regular graphs

Let us remind that a graph G is r-regular if all its vertices are of degree r. We consider the following restricted version of Partial Complement to $\mathcal{G}$.

Partial Complement to r-Regular (PCrR)
Input: A simple undirected graph G, a positive integer r.
Question: Does there exists a vertex set $S \subseteq V(G)$ such that $G \oplus S$ is r-regular?

In this section, we show that Partial Complement to r-Regular is NP-complete by a reduction from Clique in r-regular Graph.

Clique in r-regular Graph (KrR)
Input: A simple undirected graph G which is r-regular, a positive integer k.
Question: Does G contain a clique on k vertices?

We will need the following well-known proposition.

▶ Proposition 4 ([7]). Clique in r-regular Graph is NP-complete.

▶ **Theorem 11.** Partial Complement to r-Regular *is NP-complete.*

Proof. We begin by defining a gadget which we will use in the reduction. For integers $r > k$ such that $r-k$ is even, we build the graph $\text{GDG}_{k,r}$ as follows. Initially, we let $\text{GDG}_{k,r}$ consist of one clique on $k-1$ vertices, as well as $k-1$ distinct copies of $K_{r,r}$. These are all the vertices of the gadget, which is a total of $(k-1)+2r\cdot(k-1)$ vertices. We denote the

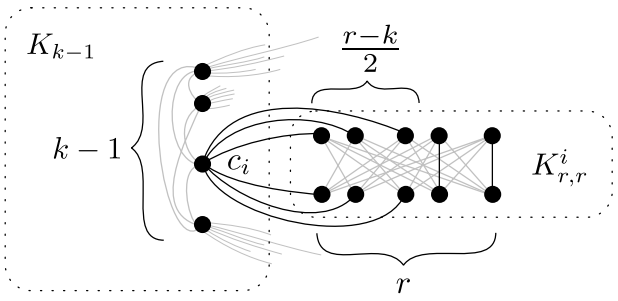

Figure 1 The graph $\text{GDG}_{k,r}$ is built of k parts, namely a clique K_{k-1}, and $k-1$ complete bipartite graphs $K^1_{r,r}, \ldots, K^{k-1}_{r,r}$ with some rewiring.

vertices of the clique $c_1, c_2, \ldots, c_{k-1}$, and we let the complete bipartite graphs be denoted by $K^1_{r,r}, K^2_{r,r}, \ldots, K^{k-1}_{r,r}$. For a bipartite graph $K^i_{r,r}$, let the vertices of the two parts be denoted by $a^i_1, a^i_2, \ldots, a^i_r$ and $b^i_1, b^i_2, \ldots, b^i_r$ respectively.

We will now do some rewiring of the edges to complete the construction of $\text{GDG}_{k,r}$. Recall that $r-k$ is even and positive. For each vertex c_i of the clique, add one edge from c_i to each of $a^i_1, a^i_2, \ldots, a^i_{\frac{r-k}{2}}$. Similarly, add an edge from c_i to each of $b^i_1, b^i_2, \ldots, b^i_{\frac{r-k}{2}}$. Now remove the edges $a^i_1 b^i_1, a^i_2 b^i_2, \ldots, a^i_{\frac{r-k}{2}} b^i_{\frac{r-k}{2}}$. Once this is done for every $i \in [k-1]$, the construction is complete. See Figure 1.

We observe the following property of vertices a^i_j, b^i_j, and c_i of $\text{GDG}_{k,r}$.

▶ Observation 2. For every $i \in [k-1]$ and $j \in [r]$, it holds that the degrees of a^i_j and b^i_j in $\text{GDG}_{k,r}$ are both exactly r, whereas the degree of c_i is $r-1$.

We are now ready to prove that Clique in r-regular Graph is many-one reducible to Partial Complement to r-Regular.

▶ Algorithm 2 (Reduction KrR to PCrR).
Input: An instance (G, k) of KrR.
Output: An instance (G', r) of PCrR such that it is a yes-instance if and only if (G, k) is a yes-instance of KrR.

1. If $k < 7$ or $k \geq r$, solve the instance of KrR by brute force. If it is a yes-instance, return a trivial yes-instance to PCrR, if it is a no-instance, return a trivial no-instance to PCrR.
2. If $r-k$ is odd, modify G by taking two copies of G which are joined by a perfect matching between corresponding vertices. Then r increase by one, whereas k remains the same.
3. Construct the graph G' by taking the disjoint union of G and the gadget $\text{GDG}_{k,r}$. Here, r denotes the regularity of G after step 2 is performed. Return (G', r).

Let $n = |V(G)|$. We observe that the number of vertices in the returned instance is at most $2n + (k-1) + 2r \cdot (k-1)$, which is $\mathcal{O}(n^2)$. The running time of the algorithm is $\mathcal{O}(n^7)$ and thus is polynomial.

The correction of the reduction follows from the following two lemmata.

▶ **Lemma 12.** *Let (G, k) be the input of Algorithm 2, and let (G', r) be the returned result. If (G, k) is a yes-instance to* Clique in r-regular Graph*, then (G', r) is a yes-instance of* Partial Complement to r-Regular.

Proof. Let $C \subseteq V(G)$ be a clique of size k in G. If the clique is found in step 1, then (G', r) is a trivial yes-instance, so the claim holds. Thus, we can assume that the graph G' was constructed in step 3. If G was altered in step 2, we let C be the clique in one of the two

copies that was created. Let $S \subseteq V(G')$ consist of the vertices of C as well as the vertices of the clique K_{k-1} of the gadget $\text{GDG}_{k,r}$. We claim that S is a valid solution to (G', r).

We show that $G' \oplus S$ is r-regular. Any vertex not in S will have the same number of neighbors as it had in G'. Since the only vertices that weren't originally of degree r were those in the clique K_{k-1}, all vertices outside S also have degree r in $G' \oplus S$. What remains is to examine the degrees of vertices of C and of K_{k-1}.

Let c_i be a vertex of K_{k-1} in G'. Then c_i lost its $k-2$ neighbors from K_{k-1}, gained k neighbors from C, and kept $r-k$ neighbors in $K^i_{r,r}$. We see that its new neighborhood has size $k + r - k = r$.

Let $u \in C$ be a vertex of the clique from G. Then u lost $k-1$ neighbors from C, gained $k-1$ neighbors from K_{k-1}, and kept $r-(k-1)$ neighbors from $G - C$. In total, u will have $r-(k-1)+(k-1) = r$ neighbors in $G' \oplus S$. Since every vertex of $G' \oplus S$ has degree r, it is r-regular, and thus (G', r) is a yes-instance. ◀

▶ **Lemma 13** (∗). *Let (G, k) be the input of Algorithm 2, and let (G', r) be the returned result. If (G', r) is a yes-instance to* Partial Complement to r-Regular, *then (G, k) is a yes-instance of* Clique in r-regular Graph.

Proof of Lemma 13 is found in the appendix. Lemmata 12 and 13 together with Proposition 4 conclude the proof of NP-hardness. Membership in NP is trivial, so NP-completeness holds. ◀

We remark that if r is a constant not given with the input, the problem becomes polynomial time solvable by Theorem 4.

7 Conclusion and open problems

In this paper we initiated the study of Partial Complement to $\mathcal{G}$. Many interesting questions remain open. In particular, what is the complexity of the problem when $\mathcal{G}$ is

- the class of chordal graphs,
- the class of interval graphs,
- the class of graph excluding a path P_5 as an induced subgraph,
- the class graphs with max degree $\leq r$, or
- the class of graphs with min degree $\geq r$

Acknowledgement We thank Saket Saurabh for helpful discussions.

References

1 Alexandre Blanché, Konrad Dabrowski, Matthew Johnson, Vadim V. Lozin, Daniël Paulusma, and Victor Zamaraev. Clique-width for graph classes closed under complementation. *CoRR*, abs/1705.07681, 2017. URL: `http://arxiv.org/abs/1705.07681`, `arXiv:1705.07681`.

2 B. Courcelle, J. A. Makowsky, and U. Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150, Apr 2000. URL: `https://doi.org/10.1007/s002249910009`, `doi:10.1007/s002249910009`.

3 Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1-3):77–114, 2000.

4 Bruno Courcelle and Sang-il Oum. Vertex-minors, monadic second-order logic, and a conjecture by seese. *J. Combinatorial Theory Ser. B*, 97(1):91–126, 2007. URL: `https://doi.org/10.1016/j.jctb.2006.04.003`, `doi:10.1016/j.jctb.2006.04.003`.

5 Professor Bruno Courcelle and Dr Joost Engelfriet. *Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach*. Cambridge University Press, New York, NY, USA, 1st edition, 2012.

6 Tomas Feder, Pavol Hell, Sulamita Klein, and Rajeev Motwani. List partitions. *SIAM Journal on Discrete Mathematics*, 16(3):449–478, 2003.

7 Michael R. Garey and David S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979.

8 Petr Hlinený and Sang-il Oum. Finding branch-decompositions and rank-decompositions. *SIAM J. Computing*, 38(3):1012–1032, 2008.

9 Marcin Kamiński, Vadim V. Lozin, and Martin Milanič. Recent developments on graphs of bounded clique-width. *Discrete Applied Mathematics*, 157(12):2747 – 2761, 2009. Second Workshop on Graph Classes, Optimization, and Width Parameters. URL: `http://www.sciencedirect.com/science/article/pii/S0166218X08003351`, `doi:http://dx.doi.org/10.1016/j.dam.2008.08.022`.

A Appendix

Clique-width

Let G be a graph and k be a positive integer. A *k-graph* is a graph whose vertices are labeled by integers from $\{1, 2, \ldots, k\}$. We call the k-graph consisting of exactly one vertex labeled by some integer from $\{1, 2, \ldots, k\}$ an initial k-graph. The *clique-width* of G, denoted by $\text{CWD}(G)$, is the smallest integer k such that G can be constructed by means of repeated application of the following four operations on k-graphs: (1) *introduce*: construction of an initial k-graph labeled by i and denoted by $i(v)$ (that is, $i(v)$ is a k-graph with v as a single vertex and label i), (2) *disjoint union* (denoted by $\dot\cup$), (3) *relabel*: changing all labels i to j (denoted by $\rho_{i\to j}$), and (4) *join*: connecting all vertices labeled by i with all vertices labeled by j by edges (denoted by $\eta_{i,j}$). Using the symbols of these operations, we can construct well-formed expressions. An expression is called *k-expression* for G if the graph produced by performing these operations, in the order defined by the expression, is isomorphic to G when labels are removed, and the clique-width of G is the minimum k such that there is a k-expression for G.

For integer k, we say that a graph class $\mathcal{G}$ is of clique-with at most k, if the clique-width of every graph in $\mathcal{G}$ is at most k. We also say that a graph class $\mathcal{G}$ is of *bounded clique-width*, if there is a k such that $\mathcal{G}$ is of clique-with at most k.

Monadic Second Order Logic.

$\mathbf{MSO}_1$ is the sublogic of $\mathbf{MSO}_2$ (Monadic Second Order Logic) without quantifications over edge subsets. More precisely, The syntax of $\mathbf{MSO}_1$ of graphs includes the logical connectives $\vee$, $\wedge$, $\neg$, $\Leftrightarrow$, $\Rightarrow$, variables for vertices, edges and sets of vertices, the quantifiers $\forall$, $\exists$ that can be applied to these variables, and the following five binary relations:

1. $u \in U$ where u is a vertex variable and U is a vertex set variable;
2. $\mathbf{inc}(d, u)$, where d is an edge variable, u is a vertex variable, and the interpretation is that the edge d is incident with the vertex u;

3. **adj**(u, v), where u and v are vertex variables and the interpretation is that u and v are adjacent;
4. equality of variables representing vertices, edges, sets of vertices, and sets of edges.

We refer [5] for more information on $\mathbf{MSO}_1$ and $\mathbf{MSO}_2$.

Missing proofs

Proof of Lemma 8. Let $\textsc{cwd}(G) = k$. To show the bound, it is more convenient to use expression trees instead of k-expressions. An *expression tree* of a graph G is a rooted tree T with nodes of four types i, $\dot\cup$, η and ρ:

- *Introduce nodes* $i(v)$ are leaves of T corresponding to initial i-graphs with vertices v labeled by i.
- *Union node* $\dot\cup$ stands for a disjoint union of graphs associated with its children.
- *Relabel node* $\rho_{i\to j}$ has one child and is associated with the k-graph obtained by applying of the relabeling operation to the graph corresponding to its child.
- *Join node* $\eta_{i,j}$ has one child and is associated with the k-graph resulting by applying the join operation to the graph corresponding to its child.
- The graph G is isomorphic to the graph associated with the root of T (with all labels removed).

The *width* of the tree T is the number of different labels appearing in T. If G is of clique-width k, then by parsing the corresponding k-expression, one can construct an expression tree of width k and, vise versa, given an expression tree of width k, it is straightforward to construct a k-expression. Throughout the proof we call the elements of $V(T)$ *nodes* to distinguish them from the vertices of G. Given a node x of an expression tree, T_x denotes the subtree of T rooted in x and the graph G_x represents the k-graph formed by T_x.

An expression tree T is *irredundant* if for any join node $\eta_{i,j}$, the vertices labeled by i and j are not adjacent in the graph associated with its child. It was shown by Courcelle and Olariu [3] that every expression tree T of G can be transformed into an irredundant expression tree T' of the same width in time linear in the size of T.

Let T be an irredundant expression tree of G with the width k rooted in r. We construct the expression tree T' for $G' = G \oplus S$ by modifying T.

Recall that the vertices of the graphs G_x for $x \in V(T)$ are labeled $1, \ldots, k$. We introduce three groups of distinct labels $\alpha_1, \ldots, \alpha_k$, $\beta_1, \ldots, \beta_k$ and $\gamma_1, \ldots, \gamma_k$. The labels $\alpha_1, \ldots, \alpha_k$ and $\beta_1, \ldots, \beta_k$ correspond the the labels $1, \ldots, k$ for the vertices in S and $V(G) \setminus S$ respectively. The labels $\gamma_1, \ldots, \gamma_k$ are auxiliary. Then for every node x of T we construct T'_x using T_x starting the process from the leaves. We denote by G'_x the k-graph corresponding to the root x of T'_x.

For every introduce node $i(v)$, we construct an introduce node $\alpha_i(v)$ if $v \in S$ and an introduce node $\beta_i(v)$ if $v \notin S$. Let x be a non-leaf node of T and assume that we already constructed the modified expression trees of the children of x.

Let x be a union node $\dot\cup$ of T and let y and z be its children.

We construct k relabel nodes ρ_{α_i,γ_i} for $i \in \{1, \ldots, k\}$ that form a path, make one end-node of the path adjacent to y in T'_y and make the other end-node denoted by y' the root of $T'_{y'}$ constructed from T'_y. Notice that in the corresponding graph $G'_{y'}$ all the vertices of S are now labeled by $\gamma_1, \ldots, \gamma_k$ instead of $\alpha_1, \ldots, \alpha_k$.

Next, we construct a union node $\dot\cup$ denoted by $x^{(1)}$ with the children y' and z. This way we construct the disjoint union of $G'_{y'}$ and G'_z.

Notice that the vertices that are labeled by the same label in G_y and G_z are not adjacent in G. Respectively, we should make the vertices of $V(G_x) \cap S$ and $V(G_y) \cap S$ with the same label adjacent in G'. We achieve it by adding k join nodes η_{α_i,γ_i} for $i \in \{1,\ldots,k\}$, forming a path out of them and making one end-node of the path adjacent to $x^{(1)}$. We declare the other end-node of the path denoted by $x^{(2)}$ the new root.

Observe now that for the set of vertices Y_i of G_y labeled i and the set of vertices Z_j of G_z labeled by j where $i,j \in \{1,\ldots,k\}$ are distinct, it holds that the vertices of Y_i and Z_j are either pairwise adjacent in G or pairwise nonadjacent. Respectively, on this stage of construction we ensure that if the vertices of Y_i are not adjacent to the vertices of Z_j, then the vertices of $Y_i \cap S$ and $Z_j \cap S$ are made adjacent in G'. To do it, for every two distinct $i,j \in \{1,\ldots,k\}$ such that the vertices of Y_i and Z_j are not adjacent in G, construct a new join node η_{γ_i,α_j} and form a path with all these nodes whose one end-node is adjacent to $x^{(2)}$ and the other end-node $x^{(3)}$ is the new root (we assume that $x^{(3)} = x^{(2)}$ if have no new constricted nodes).

Finally, we add k relabel nodes ρ_{γ_i,α_i} for $i \in \{1,\ldots,k\}$ that form a path, make one end-node of the path adjacent to $x^{(3)}$ and make the other end-node denoted by x the root of the obtained T'_x. Clearly, all the vertices of S in G'_x are labeled by $\alpha_1,\ldots,\alpha_k$.

Let x be a relabel node $\rho_{i\to j}$ of T and let y be its child. We construct two relabel nodes $\rho_{\alpha_i\to\alpha_j}$ and $\rho_{\beta_i\to\beta_j}$ denoted by x and x' respectively. We make x' the child of x and we make the root y of T'_y the child of x'.

Now, let x be a join node $\eta_{i\to j}$ of T and let y be its child. Recall that T is irredundant, that is, the vertices labeled by i and j in G_y are not adjacent. Clearly, we should avoid making adjacent the vertices in S in the construction of G'. We do it by constructing three new join nodes $\eta_{\alpha_i\to\beta_j}$, $\eta_{\alpha_j\to\beta_i}$ and $\eta_{\beta_i\to\beta_j}$ denoted by x, x', x'' respectively. We make x' the child of x, x'' the child of x' and the node y of T'_y is made the child of x''.

This completes the description of the construction of T'. Using standard inductive arguments, it is straightforward to verify that G' is isomorphic to the graph associated with the root of T', that is, $\textsc{cwd}(G') \leq 3k$. ◀

Proof of Lemma 13. Let $S \subseteq V(G')$ be a solution witnessing that (G', r) is a yes-instance. If (G', r) was the trivial yes-instance returned in step 1 of Algorithm 2, the statement trivially holds. Going forward we may thus assume (G', r) was returned in step 3, and that $k \geq 7$.

Because $G' \oplus S$ is r-regular, it must be the case that every vertex of K_{k-1} is in S, since by construction these are the vertices which do not have degree r in G'.

We claim that $|S| = 2k-1$, and moreover, that no neighbor of K_{k-1} is in S. To show this, we let $p = |S \setminus K_{k-1}|$, and proceed to show that $p = k$. Towards this end, consider a vertex $c_i \in K_{k-1}$. This vertex has some number of neighbors in $S \setminus K_{k-1}$, denoted $x_i = |N_{G'}(c_i) \cap (S \setminus K_{k-1})|$. We know that c_i has r neighbors in $G' \oplus S$. Let us count them: Some neighbors are preserved by the partial complementation, namely $r - k - x_i$ of its neighbors found in $K^i_{r,r}$. Some neighbors are gained, namely $p - x_i$ of the vertices in S. Thus, we have that $r = r - k - x_i + p - x_i$. The r's cancel, and we get $0 = p - k - 2x_i$. This is true for every $i \in [k-1]$, so we simply denote the number by $x = x_i$, and get $p = k + 2x$.

Towards the claim, it remains to show that $x = 0$. Because the neighborhoods of distinct c_i and c_j are disjoint outside K_{k-1}, we get that $p \geq (k-1) \cdot x$. We substitute p, and get

$$k + 2x \geq (k-1) \cdot x$$

$$k \geq (k-3) \cdot x$$

$$\frac{k}{k-3} \geq x$$

Recalling that $k \geq 7$, we have that x is either 1 or 0. Assume for the sake of contradiction that $x = 1$. Then without loss of generality, each c_i has some neighbor a^i_j which is in S. Since a^i_j had degree r in G', it must hold that a^i_j has equally many neighbors as non-neighbors in S. At most one of a^i_j's neighbors is outside of $K^i_{r,r}$, this means that at least $\frac{|S|-3}{2}$ vertices of $K^i_{r,r}$ are in S. Because $k \geq 7$ and the $K^i_{r,r}$'s are completely disjoint for different values of $i \in [k-1]$, we get that

$$|S| \geq \frac{|S|-3}{2} \cdot (k-1) \geq \frac{|S|-3}{2} \cdot 6$$
$$|S| \geq 3 \cdot |S| - 9$$
$$9 \geq 2 \cdot |S|$$

Seeing that $|S| \geq k-1 \geq 6$, this is a contradiction. Thus, x must be 0, so $p = k + 2x = k$ and the claim holds.

We now show that $S \setminus K_{k-1}$ is a clique in G'. Assume for the sake of contradiction it is not, and let $u, v \in S \setminus K_{k-1}$ be vertices such that $uv \notin E(G')$. Consider the vertex u. By the above claim we know that u does not have a neighbor in K_{k-1}. It will thus gain at least k edges going to $K_{k-1} \cup \{v\}$, and lose at most $k-2$ edges going to $S \setminus (K_{k-1} \cup \{u, v\})$. Because u was of degree r in G' yet gained more edges than it lost by the partial complementation, its degree is strictly greater than r in $G \oplus S$. This is a contradiction, hence $S \setminus K_{k-1}$ is a clique in G'.

Because $k \geq 3$, the clique $S \setminus K_{k-1}$ can not be contained in the gadget $\text{GDG}_{k,r}$ nor span across both copies of G created in step 2 of the reduction (if that step was applied). It must therefore be contained in the original G. Thus, G has a clique of size k, and (G, k) is a yes-instance of Clique in r-regular Graph. ◀