FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

BACHELOR THESIS

Pavel Madaj

Vertex-transitive Supergraphs

Department of Applied Mathematics

Supervisor of the bachelor thesis:	doc. RNDr. Martin Tancer, Ph.D.
Study programme:	Computer Science
Study branch:	General Computer Science

Prague 2021

I declare that I carried out this bachelor thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date .... signature of the author

This thesis is dedicated to my parents and my thesis supervisor doc. RNDr. Martin Tancer without support of which I would most definitely not be able to complete it.

Title: Vertex-transitive Supergraphs

Author: Pavel Madaj

Department: Department of Applied Mathematics

Supervisor: doc. RNDr. Martin Tancer, Ph.D., Department of Applied Mathematics

Abstract: In this thesis we explore ways to extend graphs to supergraphs that are vertex-transitive. We introduce a template system for their construction. This system is used to provide a construction of vertex-transitive supergraphs of exponential size for general graphs and of quadratic size for bipartite graphs. For general graphs we also provide a quadratic lower bound. We also sketch an approach that could lead to bridging the time complexity gap between the graph isomorphism problem and the problem of recognizing vertex-transitive graphs.

Keywords: graph theory, symmetry, automorphisms, computational complexity, graph isomorphism

Contents

Introduction

In this thesis, we study how a graph can be extended to a more symmetric larger graph. We provide several constructions for various subclasses of graphs and prove bounds on the size of the extended graph.

In order to talk more rigorously about the subject of this thesis, we require several basic definitions.

Definition 1. A graph $G = (V, E)$ is called a *vertex-transitive graph* if for every pair of vertices u, v there exists a graph automorphism $f\colon V \to V$ such that $f(u) = v$.

In the language of group theory, an equivalent and more common definition would be that a graph is *vertex-transitive* if its automorphism group acts transitively on its vertices. The action of group G on set X is *transitive*, if X is non-empty and if for each pair x, y in X there exists a g in G such that $g \cdot x = y$.

Definition 2. Graph H is a *vertex-transitive supergraph of graph G* if H is a vertex-transitive graph and G is isomorphic to an induced subgraph of H.

We focus on the construction of vertex-transitive supergraphs, usually with the goal of minimizing the size of the supergraph. Note that if we only required G to be a subgraph of H (as opposed to being an *induced* subgraph) the task would be trivial as the complete graph on the vertices of G would always satisfy this property.

In order to construct these graphs, we define a *template* which lets us take several copies of the input graph and stitch them together into a vertex-transitive supergraph. We prove that every vertex-transitive supergraph can be constructed using a template (possibly first requiring the removal of some extraneous edges). We provide a construction of a template schema, which lets us prove the following theorem.

Theorem 3. *For every graph G of order n there exists graph H such that H is a vertex-transitive supergraph of G and is of order 2^{n-1}.*

For bipartite graphs, we provide a much more compact construction which gives us a much tighter bound.

Theorem 4. *For every bipartite graph G of order n, there exists graph H such that H is a vertex-transitive supergraph of G and is of order $2n^2$.*

For general graphs, we also prove a lower bound of $(\frac{n-1}{2})^2$ for the order of vertex-transitive supergraphs. Specifically we prove that for every $n \in \mathbb{N}$ there is a graph G such that every vertex-transitive supergraph of G has at least $(\frac{n-1}{2})^2$ vertices.

In the second section, we establish how these results could be connected with complexity theory. We are interested in proving that the problem of recognizing vertex-transitive graphs is of the same difficulty as the notorious graph isomorphism problem. This is not achieved in this thesis, although a connection is made which would provide full proof if a vertex-transitive supergraph construction having certain properties is found.

All graphs in this thesis are finite, undirected, and simple unless stated otherwise.

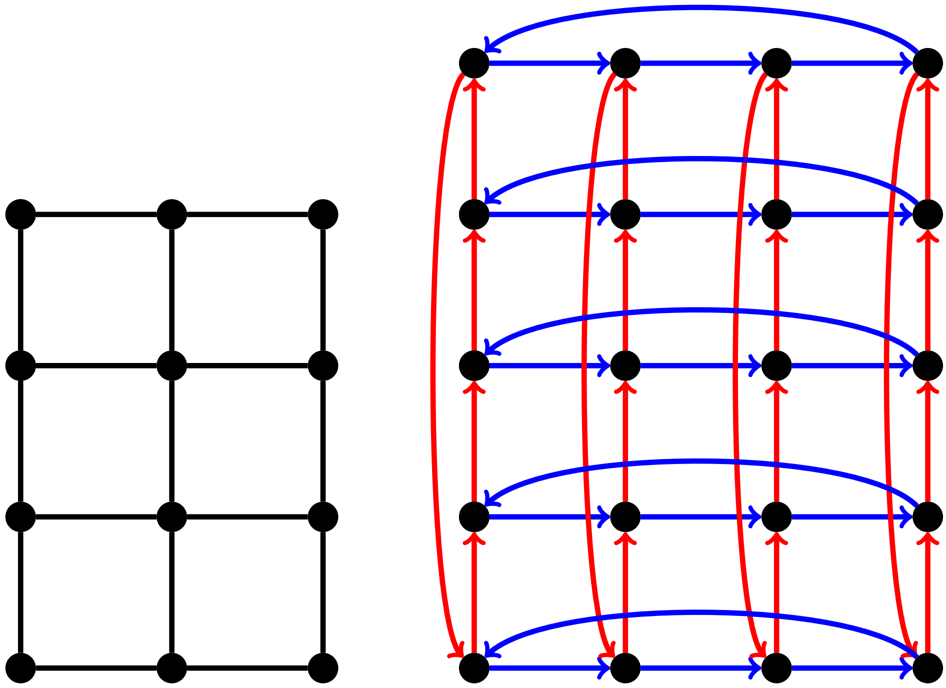

Figure 1: The grid graph $P_3 \times P_4$ on the left and the Cayley graph of the group $\mathbb{Z}_4 \times \mathbb{Z}_5$ on the right. The first graph is an induced subgraph of the undirected version of the second graph and can thus "inherit" the cardinal directions.

Motivation

The motivation behind this thesis is twofold. The original motivation stems from computational complexity, specifically the relationship of the problem of testing vertex-transitivity of a graph and the graph isomorphism problem. Construction of vertex-transitive supergraphs could lead to a possible way to prove that these two problems might be equally hard. This line of thinking is explored in greater detail in the second section.

There is also secondary, less-rigorous motivation behind this area of study. If we are given an infinite grid graph (the product of two infinite path graphs), we can naturally assign consistent cardinal directions (north, south, east and west) to each pair of a vertex and an edge incident to it. In this case, the directions can also be understood as interpreting the graph as the Cayley graph of the group $\mathbb{Z}^2$. Recognizing Cayley graphs therefore seems like a good first step for adding this structure of directions to an arbitrary graph. However, intuitively we feel that a finite grid graph (the product of two finite path graphs) should also be assigned a similar, though incomplete, structure. And yet, this graph is not even regular and as such it most certainly cannot be a Cayley graph. One way to repair this issue is to try to look at the graph as a subgraph of some larger graph that is indeed a Cayley graph and then retrieve the directional structure from this supergraph. In the case of a graph $P_m \times P_n$, this would be the Cayley graph of the group $\mathbb{Z}_{m+1} \times \mathbb{Z}_{n+1}$ (see Figure 1).

The property of being a Cayley graph is stronger than just being a vertex-transitive graph. However, the gap between these two properties is smaller than it is immediately apparent. A theorem due to Sabidussi states that vertex-transitive graphs can be understood as a generalization of Cayley graphs. Specifically in Cayley graphs, vertices correspond to group elements while in vertex-transitive graphs Sabidussi's construction uses cosets of a subgroup for vertices. [1] [7] [9] Therefore, it is not unreasonable to expect that the work done here on vertex-transitive supergraphs could be extended into a similar study of Cayley supergraphs.

Preliminaries

The topic of this thesis deals with graph theory, so we will establish some base graph theory terms. For a deeper overview, we recommend reading common graph theory introductory literature, for example [3].

Definition 5. A *(simple undirected) graph* G is a pair (V, E) where V is a set and $E \subseteq \binom{V}{2}$. We call V the set of *vertices of* G and denote it by $V(G)$, similarly we call E the set of *edges of* G and denote it by $E(G)$.

When it is clear from the context that we are talking about edges, we shall also use the notation uv for the edge $\{u, v\}$.

The *order of* G is the number of vertices (i.e. $|V(G)|$).

We say that graph H is a *subgraph of* G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.

If $X \subseteq V(G)$, we define the *subgraph induced in* G *by* X as $(X, E(G) \cap \binom{X}{2})$. We denote this graph by $G[X]$.

Now we shall define the commonly used terms about graph homomorphisms:

Definition 6. Let G and H be graphs and $f\colon V(G) \to V(H)$ a mapping between them. We say that f is a *(graph) homomorphism* if $\{f(u), f(v)\} \in E(H)$ for every edge $\{u, v\} \in E(G)$.

If f has an inverse and that inverse is also a homomorphism, then we call f a *(graph) isomorphism.*

We say that graphs G and H are *isomorphic* if there exists an isomorphism between them.

If f is an isomorphism $F\colon G \to G$, then we call it an *automorphism.*

Automorphisms of graph G form a group with the operation of composition. We denote this group by $\mathrm{Aut}(G)$ and call it the *automorphism group of* G.

[1]Given a group G, a subgroup H, and a set $D \subseteq G$, the Sabidussi graph, $\mathrm{Sab}(G, H, D)$, is the directed graph with vertex set $cos_G(H)$ and arc set $\{(Hx, Hy)\colon xy^{-1} \in D\}$. Sabidussi's theorem states that all Sabidussi graphs are vertex-transitive and all vertex-transitive graphs are Sabidussi. It is easy to see that for H being the trivial one element subgroup and D being a set of generators we get Cayley graphs.

1. Vertex-transitive Supergraph Construction

1.1 Basic Results

At a first glance, one might believe that the minimal order of a vertex-transitive supergraph of some graph G directly depends on the symmetries of G. As a counterexample, let us consider a path graph P_{k+3} with one additional vertex connected with an edge to the third vertex of the path (see Figure 1.1). For values $k \leq 3$ this graph's automorphism group is trivial, so the graph is as asymmetric as possible. And yet, we can create a relatively small vertex-transitive supergraph consisting of $2k + 8$ vertices. That gives us a ratio of 2 between the orders of the two graphs. As we will see later, achieving a constant ratio is impossible for common classes of graphs so a ratio of 2 is exceptional.

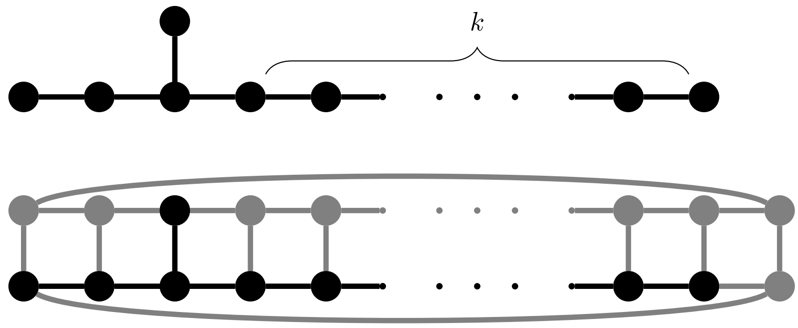

Figure 1.1: A graph with a trivial automorphism group and its vertex-transitive supergraph.

Let us establish some basic facts about the structure of vertex-transitive supergraphs. We will often talk about different copies of the base graph in its vertex-transitive supergraph. We shall now formalize this notion of a copy of one graph in another.

Definition 7. Given two graphs G and H, an *induced embedding of G into H* is a graph homomorphism $h\colon V(G) \to V(H)$ such that h restricted to its image is an isomorphism. We also denote the set of all induced embeddings of G into H by $\mathrm{IE}_{G,H}$. Sometimes we use the phrase *copy of G in H*, by this we mean either an induced embedding of G into H or its image (the actual induced subgraph of H isomorphic to G).

Observation 8. *If H is a graph and G is isomorphic to some induced subgraph of H, then there exists at least one induced embedding of G into H.*

Proof. Let $H = (V_H, E_H)$. Then G is isomorphic to $H[X]$ for some $X \subseteq V_H$. We take this isomorphism $h\colon G \to H[X]$ and extend its codomain to V_H. Restriction back to the set X (the image of h) gives us the original isomorphism. $\square$

Lemma 9. *Let $G = (V_G, E_G)$ be a graph and let $H = (V_H, E_H)$ be a vertex-transitive supergraph of G. Then for every vertex $v \in V_G$ and vertex $u \in V_H$ there exists an induced embedding h of G into H such that $h(v) = u$.*

Proof. By the definition of a vertex-transitive supergraph G is isomorphic to an induced subgraph of H. By applying Observation 8 we get an induced embedding $f: G \to H$. Now we use the fact that H is a vertex-transitive graph. If we choose vertices $f(v)$ and u there has to exist an automorphism $g: H \to H$ such that $g(f(v)) = u$. Thus the first condition is satisfied by the homomorphism $g \circ f$ and since g is an isomorphism the second property remains satisfied. □

This lemma tells us that each vertex of H plays a role of each vertex of G in some copy of G. We make extensive use of this fact throughout the thesis. It can be strengthened showing us that the structure of induced embeddings in a vertex-transitive graph is very regular:

Lemma 10. *If $G = (V_G, E_G)$ is a graph and $H = (V_H, E_H)$ is its vertex-transitive supergraph, then there exists $k \in \mathbb{N}$ such that for every $v \in V_G, u \in V_H$ there are exactly k induced embeddings $h: G \to H$ such that $h(v) = u$.*

Proof. For $v \in V_G, u \in V_H$ we define $k_{v,u}$ as the number of induced embeddings h such that $h(v) = u$. Our goal is to prove that all $k_{v,u}$ are equal.

We show that for any $x, y \in V_H, v \in V_G$ it holds that $k_{v,x} = k_{v,y}$. Since H is vertex-transitive, there exists an automorphism f of H such that $f(x) = y$. For any induced embedding h such that $h(v) = x$ the composition $f \circ h$ is an induced embedding such that $(f \circ h)(v) = y$. On the other hand, for every induced embedding h' such that $h(v) = y$ we get that $f^{-1} \circ h$ is an induced embedding such that $(f^{-1} \circ h)(v) = x$. Therefore, we have a bijection between these two sets of induced embeddings and $k_{v,x} = k_{v,y}$.

Since we have shown that for fixed v the values of $k_{v,u}$ are all equal, we denote this value by k_v. Now if we pick a fixed $v \in V_G$, then the following holds

$$|\operatorname{IE}_{G,H}| = \sum_{u \in V_H} k_{v,u} = |V_H| k_v$$

because for each induced embedding h there exists exactly one $u \in V_H$ such that $h(v) = u$. This yields that for every vertex v of G the following holds $k_v = \frac{|\operatorname{IE}_{G,H}|}{|V_H|}$. □

This simple result gives us an important insight into the structure of vertex-transitive supergraphs. If H is a vertex-transitive supergraph of G, then G occurs as an induced subgraph many times in H. There needs to exist at least one copy of G in H for each vertex of H. This observation also gives us a way to talk about constructions of vertex-transitive supergraphs. If we want to construct a vertex-transitive supergraph of G, we choose a natural number k and create a graph consisting of k copies of G. Then we take a quotient of this graph according to some partition that identifies vertices of the copies.

1.2 Templates

In this thesis, we care about the computational aspects of these constructions. We want to construct vertex-transitive supergraphs in polynomial time. An algorithm for this kind of construction could in theory analyse the structure of the input graph and then exploit repeating patterns or other structures in that graph to create a small vertex-transitive supergraph. However, this approach is unlikely to be of much use. We would need to find these possibly useful patterns quickly, but it seems that this could stray into the territory of the Induced Subgraph Isomorphism Problem, which is known to be **NP**-complete or at least the Graph Isomorphism Problem, which does not yet have a polynomial time algorithm. Moreover, we expect that for sufficiently large and sufficiently asymmetric graphs these exploitable patterns are likely absent.

With these arguments supporting us, we adopt an approach in which our construction does not care about the actual input graph. Instead, we try to provide a system that gives us a template for each order of the input graph. A universal template of order n allows us to create a vertex-transitive supergraph of any graph of order n.

Definition 11. A *template of order* n is a pair $(c, \sim)$ where $c \in \mathbb{N}$ and $\sim$ is an equivalence relation on the set $[n] \times [c]$ such that for every $k \in [c]$ there are no distinct $j_1, j_2 \in [n]$ for which $(j_1, k) \sim (j_2, k)$. (In other words, we do not want two vertices in one copy to merge into one.)

The *application of a template* $T = (c, \sim)$ *to a graph* $G = ([n], E)$ is a graph H constructed in the following way. First we create a graph G' as c disjoint copies of G. That is $G' = (V', E')$, where $V' = [n] \times [c]$ and $E' = \{(e_1, i)(e_2, i) | e_1 e_2 \in E, i \in [c]\}$). Then H is the quotient graph we get from G' by identifying vertices related by $\sim$. We are working with simple graphs so multiple parallel edges get replaced by a single edge. More explicitly $H = (V'/\sim, \{\{[u]_\sim, [v]_\sim\} | uv \in E'\})$. We denote this by $T(G)$.

We say that a template T of order n is *universal* if for every graph $G = ([n], E)$ the application of the template T to G is a vertex-transitive supergraph of G.

A template $T = (c, \sim)$ of order n is *strict* if for every pair of distinct $k, l \in [c]$ there exists at most one pair $i, j \in [n]$ such that $(i, k) \sim (j, l)$.

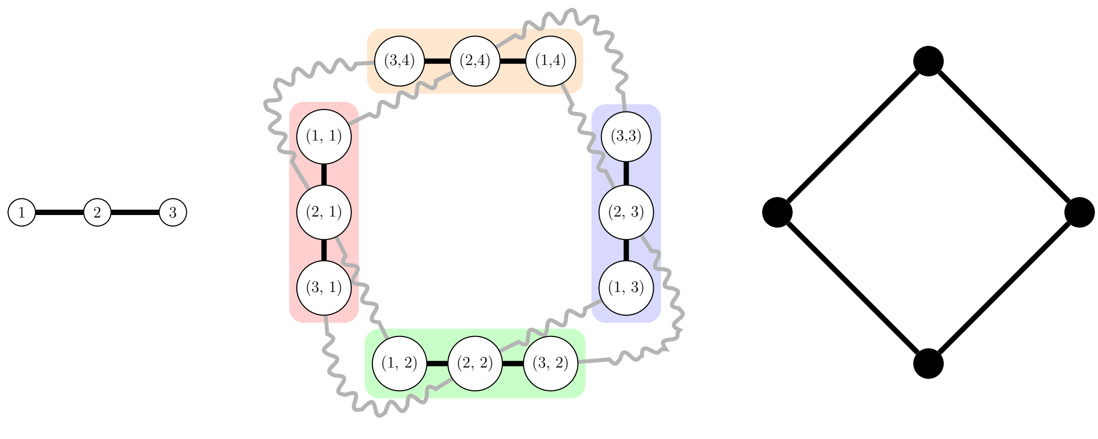

Figure 1.2: Graph P_3 on the left, a template of order 3 in the middle and the application of the template to the graph on the right. The template is drawn with 4 copies of P_3, the squiggly edges represent the $\sim$ equivalence of the template (omitting transitive edges). Note that the template is neither strict nor universal.

For an example of a template and its application see Figures 1.2 and 1.3. Alternatively we can also describe the application equivalently as: The

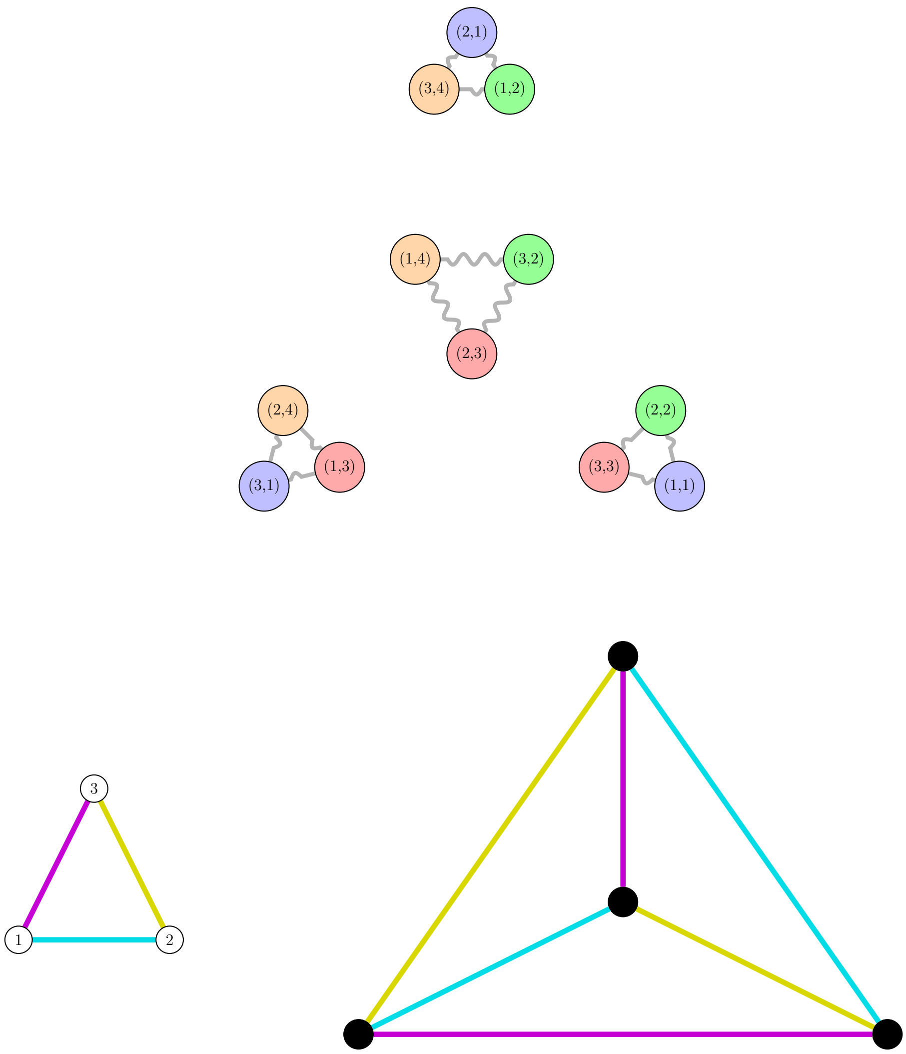

Figure 1.3: A universal template of order 3 in the top row. A graph G in the bottom row on the left with application of the template to G on the right. Corresponding edges in these two graphs have the same colour.

application of a template $T = (c, \sim)$ to a graph $G = ([n], E)$ is a graph $H = (([n] \times [c])/\sim, E_H)$, where XY is an edge of H if and only if there exist $k \in [c]$ and $u, v \in V_G$ such that $(u, k) \in X$ and $(v, k) \in Y$. This alternative definition also immediately gives us the following result that we use throughout the next section.

Observation 12. *If G is a graph of order n, $T = (c, \sim)$ is a template of order n, and uv is an edge in $T(G)$, then there exist $i, j \in V(G)$ and $k \in [c]$ such that $u = [(i, k)]_\sim$ and $v = [(j, k)]_\sim$.*

We also invoke the application of a template on a graph with a different vertex set. In that case, let us without loss of generality replace that graph with an isomorphic graph with the vertex set $[n]$.

The motivation behind the definition of a strict template is that if a template is strict, then each edge lies in exactly one copy of G. This guarantees us that G is isomorphic to an induced subgraph of the application of the template to G.

Observation 13. *If $T = (c, \sim)$ is a strict template of order n and $G = ([n], E)$ is a graph then G is always isomorphic to an induced subgraph of $T(G)$.*

Proof. We claim that $u \mapsto [(u, 1)]_\sim$ gives us the isomorphism between G and the subgraph induced by the image of that mapping.

By the condition in the definition of a template, we know that if we have a pair $i, j \in [n]$ such that $(i, 1) \sim (j, 1)$, then necessarily $i = j$. So no two of the vertices from the first copy lie in the same equivalence class of $\sim$ and therefore the mapping is injective.

Let $H = T(G)$. Then uv is an edge in H if and only if $u = [(i, k)]_\sim$, $v = [(j, k)]_\sim$ for some $k \in [c]$ and ij is an edge in G. As a direct consequence we get that $ij \in E(G) \implies \{[(i, 1)]_\sim, [(j, 1)]_\sim\} \in E(H)$. To prove the other direction, let us assume for contradiction that $ij \notin E(G)$ and $\{[(i, 1)]_\sim, [(j, 1)]_\sim\} \in E(H)$. By Observation 12 there exists $i'j' \in E$ and $k' \in [c]$ such that $(i, 1) \sim (i', k')$ and $(j, 1) \sim (j', k')$. Now there are two cases:

$k' = 1$: Because $ij \notin E(G)$ either $i \neq i'$ or $j \neq j'$, without loss of generality let it be the first case. But then $(i, 1) \sim (i', 1)$ and $i \neq i'$, which contradicts the fact that T is a template.

$k' \neq 1$: This directly contradicts the strictness of T.

Thus, G is isomorphic to the image of the mapping defined above. □

Thanks to this observation, we only need to ensure that the template guarantees us a vertex-transitive graph. The definition of a template is motivated by the Lemma 9 in the sense that we want every vertex-transitive supergraph of G to be constructible by applying some template to G. Unfortunately, this statement does not hold as it can happen that the supergraph has extra edges that are unrelated to the copies of G as can be seen in Figure 1.4. As an example of this, consider that the prism graph with $2n$ vertices ($C_n \times P_2$) is a vertex-transitive supergraph of the cycle graph C_n. For odd n the edges connecting the two cycles in the prism are part of no copy of C_n. But not all hope is lost, if we removed those problematic edges we would be left with two disjoint copies of C_n. That graph is clearly still a vertex-transitive supergraph of C_n and is trivially constructible by a template. We shall formalize this in the following lemma.

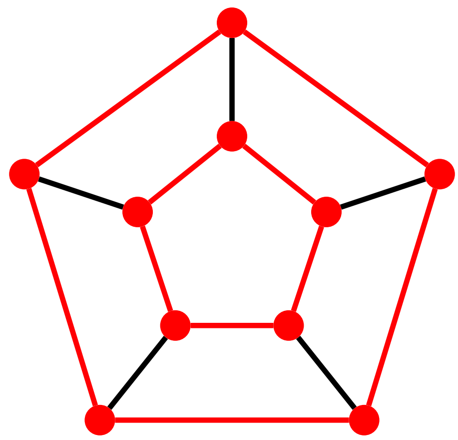

Figure 1.4: A prism graph with 10 vertices and two copies of C_5 as its induced subgraphs in red. It can be seen that the black edges do not belong to any other induced C_5 subgraph.

Theorem 14. *Let $G = (V, E)$ be a graph. Then for any graph H that is a vertex-transitive supergraph of G there exists a template $T = (c, \sim)$ and H' a subgraph of H such that $T(G) \cong H'$. The graph H' is also a vertex-transitive supergraph of G.*

Proof. We choose $c = |\mathrm{IE}_{\{} G, H\}|$. Let $\mathrm{IE}_{\{} G, H\} = \{h_1, h_2, \ldots h_c\}$ and also without loss of generality assume that $V(G) = [n]$.

We define the equivalence $\sim$ accordingly: For every $i_1, i_2 \in [n]$ and $k_1, k_2 \in [c]$ let $(i_1, k_1) \sim (i_2, k_2)$ if and only if $h_{k_1}(i_1) = h_{k_2}(i_2)$.

The condition we require of $\sim$ is satisfied because $(i_1, k) \sim (i_2, k) \iff h_k(i_1) = h_k(i_2) \iff i_1 = i_2$. The last equivalence follows from the fact that h_k is an induced embedding and therefore injective. Now $T = (c, \sim)$ is a valid template of order n and we can apply it to get $H'' = T(G)$.

We now define a mapping $f\colon V(H'') \to V(H)$ and prove that it is a homomorphism and an injection via $f([(i, k)]_\sim) = h_k(i)$. This is a sound definition because if $(i_1, k_1) \sim (i_2, k_2)$ then by definition $h_{k_1}(i_1) = h_{k_2}(i_2)$ and therefore also $f([(i_1, k_1)]_\sim) = f([(i_2, k_2)]_\sim)$.

Now we show that f is a homomorphism: Let $[(i_1, k)]_\sim[(i_2, k)]_\sim \in E(H'')$. Then $i_1 i_2 \in E$ again by the definition of template application. We know that h_k is an induced embedding and thus also a graph homomorphism, therefore $h_k(i_1)h_k(i_2) \in E(H)$. And $h_k(i) = f([(i, k)]_\sim)$, thus f is also a homomorphism.

Next we show that f is injective: If $[(i_1, k_1)]_\sim, [(i_2, k_2)]_\sim$ are two distinct vertices of H'' then $(i_1, k_1) \not\sim (i_2, k_2)$ which implies that:

$$f([(i_1, k_1)]_\sim) = h_{k_1}(i_1) \neq h_{k_2}(i_2) = f([(i_2, k_2)]_\sim)$$

Since f is an injective homomorphism, we can choose H' as the image of H'' under f (selecting only the vertices and edges to which some vertex or edge of H'' maps). When we restrict f's image to H', it becomes an isomorphism. Note that f is also surjective on the vertices due to Lemma 9.

We prove that G is isomorphic to some induced subgraph of H'' (and thus also H' which is isomorphic). Let us define $h : V(G) \to V(H'')$ as follows: $h(i) = [(i, 1)]_\sim$. By the definition of template application, we know that h is an injective graph homomorphism and thus G is isomorphic to a subgraph of H''. Let us prove that it is an induced subgraph by contradiction. Let us assume that there exists $\{u, v\} \notin E(G)$ such that $\{h(u), h(v)\} \in E(H'')$. Then there need to exist $k \in [c]$ and $\{u', v'\} \in E(G)$ such that $h(u) = [(u', k)]_\sim$ and $h(v) = [(v', k)]_\sim$. But by the definition of $\sim$ this means that $h_1(u) = h_k(u')$ and $h_1(v) = h_k(v')$. Both h_1 and h_k are induced embeddings of G into H that share a pair of vertices on their images, but h_k maps an edge to these vertices and h_1 does not. This is a contradiction, and therefore G is isomorphic to an induced subgraph of H'' and H'.

Now we shall prove that H' is vertex-transitive. Since H' and H'' are isomorphic, we instead prove this property for H''. Let $v_1 = [(i_1, k_1)]_\sim$, $v_2 = [(i_2, k_2)]_\sim$ be two distinct vertices of H''. Then by vertex-transitivity of H there exists $g\colon V(H) \to V(H)$ an automorphism of H such that $g(f(v_1)) = f(v_2)$. Given any induced embedding h_k the composition $g \circ h_k$ is also an induced embedding so it is equal to h_i for some $i \in [c]$. Note that this induces a permutation on the set $\mathrm{IE}_{\{} G, H\}$, we shall denote this permutation by π. By its definition it holds that $g \circ h_k = h_{\pi(k)}$.

We use this permutation to define a mapping $g'\colon V(H'') \to V(H'')$ and prove that it is an automorphism of H'' and that $g'(v_1) = v_2$. Let $g'([(i,k)]_\sim) = [(i,\pi(k))]_\sim$.

First we prove that $g'(v_1) = v_2$:

$$g'(v_1) = g'([(i_1,k_1)]_\sim) = [(i_1,\pi(k_1)]_\sim$$
$$g'(v_1) = v_2 \iff [(i_1,\pi(k_1)]_\sim = [(i_2,k_2)]_\sim \iff h_{\pi(k_1)}(i_1) = h_{k_2}(i_2)$$
$$h_{\pi(k_1)}(i_1) = g(h_{k_1}(i_1)) = g(f([(i_1,k_1)]_\sim)) = g(f(v_1)) = f(v_2) =$$
$$= f([(i_2,k_2)]_\sim) = h_{k_2}(i_2)$$

Now we prove that g' is a homomorphism, let $u_1u_2 \in E(H'')$ where $u_1 = [(j_1,l)]_\sim, u_2 = [(j_2,l)]_\sim$. By the construction of H'', if u_1u_2 is an edge in H'' then j_1j_2 must be an edge in G. And conversely, if j_1j_2 is an edge in G, then $[(j_1,\pi(l))]_\sim, [j_2,\pi(l)]_\sim = g'(u_1), g'(u_2)$ is an edge in H''.

Finally, we prove that g' is injective (and therefore also bijective). Due to Lemma 9 for each $v \in V(H)$ there are $i \in [n], k \in [c]$ such that $v = h_k(i)$. For such vertex it holds that:

$$f(g'(f^{-1}(v))) = f(g'([(i,k)]_\sim)) = h_{\pi(k)}(i) = g(v)$$

And since f, f^{-1} and g are all bijections g' must also be a bijection which completes our proof that g' is an automorphism of H''. □

Due to this lemma, we will from now on assume that all vertex-transitive supergraphs we encounter can be constructed by applying a template. We can do this because we generally care about minimizing the order (and possibly size) of the supergraph and the previous lemma says that if the current supergraph is not constructible by a template, then some subgraph of it is.

We would like to give a way to construct a small vertex-transitive supergraph of any graph G. However, a template of order n can only be applied to graphs of order n. To give a construction applicable to any graph we will need a template of order n for each $n \in \mathbb{N}$.

Definition 15. A sequence $\mathcal{T} = (T_i)_{i=1}^{\infty}$ is a *template schema* if T_i is a template of order i for every $i \in \mathbb{N}$. The *application of* $\mathcal{T}$ *to a graph* $G = ([n], E)$ is defined as $T_n(G)$, we denote the application by $\mathcal{T}(G)$.

We say that a template scheme $\mathcal{T} = (T_i)_{i=1}^{\infty}$ is *universal* if T_i is universal for every i.

Likewise, we say that it is strict if T_i is strict for all i.

Our goal is to provide a universal template schema that constructs reasonably small vertex-transitive supergraphs for every graph. To determine how good a template is we need some way to measure this quantity. Note that the order of the graph constructed by applying a template does not depend on the edges of the input graph, because it is equal to the number of equivalence classes of $\sim$. We assign the sequence of the orders of the constructed graphs to each template schema. In general, we care about the asymptotic behaviour of this sequence. For example if we mention a template schema $\mathcal{T} = ((c_i, \sim_i))_{i=1}^{\infty}$ produces graphs of quadratic order we mean that the sequence $(|([i] \times [c])/\sim_i|)_{i=1}^{\infty} \in \mathcal{O}(n^2)$.

1.3 Construction for General Graphs

We have not yet been successful in providing a universal template schema that would produce graphs of polynomial order. But as a partial result, let us provide an exponential template schema. In the definition of a template we have used the set $[c]$ to index the copies of the input graph. This simplifies the definitions, but can become unwieldy. In the following section, we commonly use different index sets, we implicitly assume that a bijection with the set $[c]$ is used when required.

We now describe the template $T_n = (2^{n-1}, \sim)$ of order n. As mentioned, we replace the index set $[2^{n-1}]$ with the set $C = \{X \subseteq [n] \colon |X| \text{ is even}\}$. Then $(v_1, X_1) \sim (v_2, X_2)$ if and only if $X_1 \triangle X_2 = \{v_1, v_2\}$.

Observation 16. *Let T_n be defined as above, let G be a graph of order n and let $H = T_n(G)$. Then $|H| = 2^{n-1}$.*

Proof. Each vertex of H corresponds to an equivalence class of $\sim$. Let us define $l \colon V(H) \to \{A \subseteq [n] : |A| \text{ is odd}\}$ in the following way: If $u \in V(H)$ and $(v, X) \in u$ then let $l(u) = X_1 \triangle \{v_1\}$. We prove that this is a bijection with the (2^{n-1})-element set.

We claim that this definition is sound. Let us have $(v_1, X_1), (v_2, X_2) \in u$, then by the definition of $\sim$ we have $X_1 \triangle X_2 = \{v_1, v_2\} = \{v_1\} \triangle \{v_2\}$. By rearranging this equality we get $X_1 \triangle \{v_1\} = X_2 \triangle \{v_2\}$.

We prove injectivity of l similarly: Let us have $l(u_1) = l(u_2)$, then there exist $(v_1, X_1) \in u_1$ and $(v_2, X_2) \in u_2$ such that $X_1 \triangle \{v_1\} = l(u_1) = l(u_2) = X_2 \triangle \{v_2\}$. This gives us that $X_1 \triangle X_2 = \{v_1, v_2\}$ which once again proves $(v_1, X_1) \sim (v_2, X_2)$ and therefore $u_1 = u_2$.

The set C was defined to only contain subsets of $[n]$ which are of even size so $X_1 \triangle \{v_1\}$ is always odd. To finally prove that l is surjective let us have $A \subseteq [n]$ of odd size. Then For any $v \in [n]$ we have $l([v, A \triangle \{v\}]) = A$. This concludes the proof as l provides a bijection of $V(H)$ with a 2^{n-1}-element set. $\square$

We use the mapping l to label the vertices of applications of T_n. This gives us an interesting duality between the index set C containing all even subsets of $[n]$ and the codomain of l containing all odd subsets of $[n]$. One way to think about is duality is to imagine both vertices of the template application and the copies of G indexed by C as vertices of an n-dimensional hypercube. We will not give an explicit proof, but the automorphisms we use in the following theorem are reflections of this hypercube.

Theorem 17. *The template schema $\mathcal{T} = (T_n)_{n=1}^{\infty}$ with T_n as above is universal.*

Proof. We prove that T_n is universal for each $n \in \mathbb{N}$. We start with the proof of vertex-transitivity.

Let $n \in \mathbb{N}$, $G = ([n], E)$ a graph, $H = T_n(G)$ and $u_1, u_2 \in V(H)$. We claim that H is vertex-transitive. We define a mapping $f \colon V(H) \to V(H)$ such that $f(v) = l^{-1}(l(v) \triangle l(u_1) \triangle l(u_2))$. Note that both $|l(u_1)|$ and $|l(u_2)|$ are odd so $|l(v) \triangle l(u_1) \triangle l(u_2)|$ is also odd and the application of l^{-1} is valid. We know that f is a bijection, because it is its own inverse as the symmetric differences cancel out for $f(f(v))$:

$$f(f(v)) =$$
$$l^{-1}(l(v)\triangle l(u_1)\triangle l(u_2)\triangle l(u_1)\triangle l(u_1)\triangle l(u_2)\triangle l(u_2)\triangle l(u_1)\triangle l(u_2)) =$$
$$l^{-1}(l(v)) = v$$

Finally $f(u_1) = l^{-1}(l(u_1)\triangle l(u_1)\triangle l(u_2)) = u_2$. In order to show that f is an automorphism, we also define a permutation $g\colon C \to C$ in a similar way $g(X) = X\triangle l(u_1)\triangle l(u_2)$. Now if $uv \in E(H)$, then there exists $j \in C$ such that $u = [(u', j)]_\sim, v = [(v', j)]_\sim$ and $u'v' \in E(G)$. Then also $\{[(u', g(j))]_\sim, [(v', g(j))]_\sim\} \in E(H)$.

$$l(f(u)) = l(u)\triangle l(u_1)\triangle l(u_2) = l([(u', j)]_\sim)\triangle l(u_1)\triangle l(u_2) =$$
$$= \{u'\}\triangle j\triangle l(u_1)\triangle l(u_2) = \{u'\}\triangle g(j) = l([(u', g(j))]_\sim)$$

The same holds for v and v' so if uv is an edge then $f(u)f(v)$ is also an edge and thus f is an automorphism.

We define $h\colon G \to H$ as $h(v) = [(v, \emptyset)]_\sim$. This is a graph homomorphism by the construction of H. For contradiction let $uv \in E(H)$, $u = [(u', \emptyset)]_\sim$, $v = [(v', \emptyset)]_\sim$ and $u'v' \notin E(G)$. Then there have to exist $u''v'' \in E(G), j \in C$ such that $u = [(u'', j)]_\sim$ and $v = [(v'', j)]_\sim$. We know that $j \neq \emptyset$, $(u'', j) \sim (u', \emptyset)$ and $(v'', j) \sim (v', \emptyset)$. By the construction of T_n this means that $j\triangle\emptyset = \{u', u''\} = \{v', v''\}$. From this we get that $u' = v''$ and $v' = u''$, which is a contradiction because $u''v'' \in E(G)$ and $u'v' \notin E(G)$. So G is an induced subgraph of H. □

As a corollary, we shall now prove Theorem 3.

Theorem 3. *For every graph G of order n, there exists graph H such that H is a vertex-transitive supergraph of G and is of order 2^{n-1}.*

Proof. Let us be given a graph G of order n. By Theorem 17 we know that the template schema $\mathcal{T} = (T_n)_{n=1}^\infty$ defined as above is a universal template schema. Therefore, by definition of a universal template schema $\mathcal{T}(G)$ is a vertex-transitive supergraph of G.

By Observation 16 $\mathcal{T}(G) = T_n(G)$ is of order 2^{n-1}. □

1.4 Lower Bound for General Graphs

The construction we provided creates graphs of exponential size, which is much larger than what we would hope for. If we are not able to provide a more compact solution, we will at least give a bound on how low we can go.

Theorem 18. *For every $n \in \mathbb{N}$ there exists a graph G of order n such that for every vertex-transitive supergraph H of G it holds that $|H| \geq (\frac{n-1}{2})^2$*

Proof. Let us be given n. For now we will only consider even n and let $N = \frac{n}{2}$. Our graph G will be constructed as the disjoint union of graphs $G_1 = K_N$ and $G_2 = \{[N], \emptyset\}$. Note that this means that H is also a vertex-transitive supergraph of G_1 and G_2. We shall use this fact and work with induced embeddings of G_1 and

G_2 directly instead of explicitly stating that we are working through the proxy of induced embeddings of G.

Let H be the vertex-transitive supergraph of G with the least number of vertices. We will count $|\operatorname{IE}_{G,H}|$ in two ways.

Let G_1' be an induced subgraph of H isomorphic to G_1. Then by Lemma 10 for every $u \in V(G_1')$ and $v \in V(G_2)$ there exist exactly k induced embeddings $h\colon G_2 \to H$ such that $h(v) = u$. If we do this for all choices of u and v, we do not count any induced embedding twice because G_1 and G_2 can share at most one vertex due to their opposite nature. This gives us a lower bound of $|\operatorname{IE}_{G,H}| \geq |V(G_1')||V(G_2)|k = N^2k$.

If we fix $v \in V(G)$, Lemma 10 tells us that for every $u \in V(H)$ there are exactly k induced embeddings h such that $h(v) = u$. Clearly, this counts every induced embedding exactly once as they all map v to exactly one vertex. So we get $|\operatorname{IE}_{G,H}| = k|H|$.

Combining these two results we get:

$$k|H| \geq N^2k$$
$$|H| \geq N^2$$
$$|H| \geq \left(\frac{n}{2}\right)^2$$

For odd n we remove one vertex and proceed via the same argument which now gives us $|H| \geq (\frac{n-1}{2})^2$. □

1.5 Bipartite Graphs

Let us now redirect our attention to the class of bipartite graphs, where our results are more exciting. In this section, we show a template schema that is universal for bipartite graphs. It can be intuitively understood that finding smaller universal schemata should be simpler for bipartite graphs than for the general case. This is because each bipartite graph has a large independent set, so two copies of the input graph can have a large overlap which should lead to a more compact packing of them. In order to make use of this fact, we need a slightly modified version of the template defined above as we need to restrict the application, so it does not mix the two graph parts together.

Definition 19. A *bipartite template of order n* is a pair $(c, \sim)$ where $c \in \mathbb{N}$ and $\sim$ is an equivalence relation on the set $\{-1, 1\} \times [n] \times [c]$, such that for every $k \in [c]$ there are no distinct $j_1, j_2 \in [n], s_1, s_2 \in \{-1, 1\}$ for which $(s_1, j_1, k) \sim (s_2, j_2, k)$.

The *application of a bipartite template $T = (c, \sim)$ to a bipartite graph G* with parts $A = \{-1\} \times \{1, 2, \ldots, k\}$ and $B = \{1\} \times \{k+1, k+2, \ldots, n\}$ is a graph H constructed in the following way. First we create a graph G' such that $V(G') = \{-1, 1\} \times [n]$ and $(s_1, v_1)(s_2, v_2) \in E(G')$ if and only if either $(s_1, v_1)(s_2, v_2) \in E(G)$ or $(-s_1, v_1)(-s_2, v_2) \in E(G)$. Then we create a graph G'' as c disjoint copies of G'. That is $G'' = (V'', E'')$ where $V'' = V(G') \times [c]$ and $E' = \{(e_1, i)(e_2, i) | e_1e_2 \in E(G') \wedge i \in [c]\})$. Then H is the quotient graph we get from G'' by identifying vertices related by $\sim$. Or explicitly $H = (V'/\sim, \{([u]_\sim, [v]_\sim) | uv \in E'\})$. We denote this by $T(G)$.

We say that a bipartite template T of order n is *universal* if for every bipartite graph G with parts A, B such that $|A| = |B| = n$ the application of the template T to G is a vertex-transitive supergraph of G.

A bipartite template $T = (c, \sim)$ of order n is *strict* if for every distinct $k_1, k_2 \in [c]$ there exists $s \in \{-1, 1\}$ such that whenever $(s_1, i_1, k_1) \sim (s_2, i_2, k_2)$, then $s_1 = s_2 = s$.

It can be easily seen that each bipartite template of order n can be interpreted as a template of order $2n$ by a bijection between $\{-1, 1\} \times [n]$ and $[2n]$. Once again, we adopt the convention that if G has a different vertex set then we replace it with an isomorphic graph with the required vertex set (note that we require one part be of the form of $(-1, \cdot)$ and the other of the form $(1, \cdot)$). Note that $T(G)$ is also a bipartite graph with a similar way of recognizing the parts. If $[(s, i, (l, r)]_\sim$ is a vertex, it is in the left part if $s = 1$, otherwise (when $s = -1$), it is in the right part. The motivation behind strictness is again that it guarantees us that G is an induced subgraph of $T(G)$. In a strict bipartite template two copies of G can either overlap by their left parts or their right parts but not both.

Theorem 20. *Let $T = (c, \sim)$ be a strict bipartite template of order n and G a bipartite graph with parts $A = \{-1\} \times \{1, 2, \ldots, k\}$ and $B = \{1\} \times \{k+1, k+2, \ldots, n\}$, then G is always isomorphic to an induced subgraph of $T(G)$.*

Proof. The graph G' from the definition of the application of a bipartite template is just two disjoin copies of G so G is clearly an induced subgraph of G'. We claim that $(s, u) \mapsto [(s, u, 1)]_\sim$ gives us the isomorphism between G' and the subgraph induced by the image of that mapping.

By the condition in the definition of a bipartite template, we know that for any $i_1, i_2 \in [n]$ and $s_1, s_2 \in \{-1, 1\}$ if $(s_1, i_1, 1) \sim (s_2, i_2, 1)$ then i_1 and i_2 must necessarily be equal. So no two of the vertices from the first copy lie in the same equivalence class of $\sim$ and therefore the mapping is injective.

Let $H = T(G)$. Then $v_1 v_2$ is an edge in H if and only if $v_1 = [(s_1, i_1, k)]_\sim$, $v_2 = [(s_2, i_2, k)]_\sim$ for some $k \in [c]$ and $(s_1, i_1)(s_2, i_2)$ is an edge in G'. As a direct consequence we get that $(s_1, i_1)(s_2, i_2) \in E(G') \implies \{[(s_1, i_1, 1)]_\sim, [(s_2, i_2, 1)]_\sim\} \in E(H)$. To prove the other direction, let us assume for contradiction that
$(s_1, i_1)(s_2, i_2) \notin E(G')$ and $\{[(s_1, i_1, 1)]_\sim, [(s_2, i_2, 1)]_\sim\} \in E(H)$. By Observation 12 there exist $(s'_1, i'_1)(s'_2, i'_2) \in E(G')$ and $k' \in [c]$ such that $(s_1, i_1, 1) \sim (s'_1, i'_1, k')$ and $(s_1, i_1, 1) \sim (s'_2, i'_2, k')$. The graph G' is bipartite with one part of the form $(1, \cdot)$ and the other of the form $(-1, \cdot)$ so $s'_1 \neq s'_2$. Now there are two cases:

$k' = 1$: Because $(s_1, i_1)(s_2, i_2) \notin E(G)$ either $(s_1, i_1) \neq (s'_1, i'_1)$ or $(s_2, i_2) \neq (s'_2, i'_2)$, without loss of generality let it be the first case. But then $(s_1, i_1, 1) \sim (s'_1, i'_1, 1)$ and $(s_1, i_1) \neq (s'_1, i'_1)$ which contradicts the fact that T is a template.

$k' \neq 1$: Then $(s_1, i_1, 1) \sim (s'_1, i'_1, k')$ and $(s_2, i_2, 1) \sim (s'_2, i'_2, k')$ for $s_2 \neq s_1$. But the definition of a strict bipartite template mandates that $s_1 = s_2 = s'_1 = s'_2$ which is a contradiction.

Thus, G'' is isomorphic to the image of the mapping defined above which is an induced subgraph of H. □

Analogously to the definition of a template schema we define a bipartite template schema.

Definition 21. A sequence $\mathcal{T} = (T_i)_{i=1}^{\infty}$ is a *bipartite template schema* if T_i is a template of order i for every $i \in \mathbb{N}$. The *application of $\mathcal{T}$ to a bipartite graph G* with n vertices is defined as $T_n(G)$, we denote the application by $\mathcal{T}(G)$.

We say that a bipartite template scheme $\mathcal{T} = (T_i)_{i=1}^{\infty}$ is *universal* if T_i is universal for all i.

Likewise, we say that it is strict if T_i is strict for all i.

In the rest of the section, we describe a bipartite template schema that produces graphs of quadratic size. We denote this schema by $\mathcal{T} = (T_n)_{i=n}^{\infty}$, we now proceed to define $T_n = (n^2, \sim)$. As previously we describe the template using the index set $C = \mathbb{Z}_n^2$ instead of $[n^2]$. The definition of $\sim$ follows.

$$(1, i_1, (l, r_1)) \sim (1, i_2, (l, r_2)) \iff r_1 + i_2 \equiv r_2 + i_1 \pmod n$$
$$(-1, i_1, (l_1, r)) \sim (-1, i_2, (l_2, r)) \iff l_1 + i_2 \equiv l_2 + i_2 \pmod n$$

Pairs of elements not matching the previous declaration are not related by $\sim$. For an example usage of this template refer to Figure 1.5.

Theorem 22. *The bipartite template schema $\mathcal{T}$ defined above is a universal bipartite template schema.*

Proof. Let G be a bipartite graph with parts $A = \{1\} \times \{1, 2, \ldots, k\}$ and $B = \{-1\} \times \{k+1, k+2, \ldots, n\}$ and $H = \mathcal{T}(G) = T_n(G)$.

First we show that the template schema is strict. Let us have two distinct pairs $(l_1, r_1), (l_2, r_2) \in C$. If neither $l_1 = l_2$ nor $r_1 = r_2$ then there's no pair $(s_1, i_1, (l_1, r_1)) \sim (s_2, i_2, (l_2, r_2))$ according to the definition. Otherwise, either $l_1 = l_2$ in which case the first part of the definition and $s_1 = s_2 = 1$ or $r_1 = r_2$ in which case similarly $s_1 = s_2 = -1$.

By Theorem 20 G is isomorphic to some induced subgraph of H. In order to prove that H is vertex-transitive we first define several mappings and prove that they are automorphisms.

The first automorphism swaps the parts of the bipartite graph H. We define $f\colon V(H) \to V(H)$ as $f([(s, i, (l, r))]_\sim) = [(-s, i, (l, r)]_\sim$. It is easy to see that f is well-defined and a bijection (as it is its own inverse). Furthermore, it follows from the definition of bipartite template application that $(s_1, i_1)(s_2, i_2)$ is an edge in G' if and only if $(-s_1, i_1)(-s_2, i_2)$ is an edge. Since f only does this operation on all the copies of G', it is an automorphism.

The second automorphism is $g\colon V(H) \to V(H)$ defined as follows:

$$g([(s, i, (l, r)]_\sim) = [(s, i, (l+1, r))]_\sim$$

It can be easily seen that g is well-defined. Moreover, $g^n(x) = x$ so g has to be a bijection. And as g only permutes the copies of G' it is a homomorphism. The formal proof follows exactly the same pattern as many already presented, so we omit it here.

Now let $u_1, u_2 \in V(H)$ and $u_1 = [(s_1, i_1, (l_1, r_1))]_\sim, u_2 = [(s_2, i_2, (l_2, r_2))]_\sim$. We construct an automorphism h as a composition of f and g such that $h(u_1) = u_2$.

First let $h_1 = g^{i_2 - i_1} f \circ g^{l_2 - l_1}$. Then $h_2 = h_1 \circ f$ if $s_1 = -1$ and $h_2 = h_1$ otherwise. Finally, $h = f \circ h_2$ if $s_2 = 1$ and $h = h_2$ otherwise.

From now on without loss of generality $s_1 = 1$ and $s_2 = -1$ and $h_1 = h$, if that is not the case then the construction of h and h_2 fixes this.

$$\begin{gathered} h(u_1) = h([(1, i_1, (l_1, r_1))]_\sim) = g^{i_2 - i_1}([(-1, i_1, (l_2, r_1))]_\sim) = \\ = [(-1, i_1, (l_2 + i_2 - i_1, r_1))]_\sim = [(-1, i_1 + i_2 - i_1, (l_2, r_1))]_\sim = \\ = [(-1, i_2, (l_2, r_1))]_\sim = u_2 \end{gathered}$$

□

As a corollary, we shall now prove Theorem 4.

Theorem 4. *For every bipartite graph G of order n there exists graph H such that H is a vertex-transitive supergraph of G and is of order $2n^2$.*

Proof. Let us be given a bipartite graph G of order n. By Theorem 22, we know that the template schema $\mathcal{T} = (T_n)_{n=1}^{\infty}$ defined as above is a universal template schema. Therefore, by definition of a universal template schema $\mathcal{T}(G)$ is a vertex-transitive supergraph of G.

Next, we prove that $\mathcal{T}(G) = T_n(G)$ is of order $2n^2$. Vertices of this graph are the equivalence classes of $\sim$. This equivalence relation is defined on the $(2n^3)$-element set $A = \{-1, 1\} \times [n] \times \mathbb{Z}_n^2$ so we only need to proof that each equivalence class contains n elements. Let us have $(1, i_1, (l_1, r_1)) \in A$. Then by the definition of $\sim$ we have $(s, i_2, (l_2, r_2)) \sim (1, i_1, (l_1, r_1))$ if and only if $s = 1$, $l_1 = l_2$, and $r_1 + i_2 \equiv r_2 + i_1 \pmod{n}$. Clearly this linear congruence equation has exactly n solutions because r_1, i_1 are fixed and given any $i_2 \in [n]$ we can choose $r_2 = r_1 + i_2 - i_2$. The case for $s = -1$ is analogous. This proves that the equivalence class has n elements which completes the proof. □

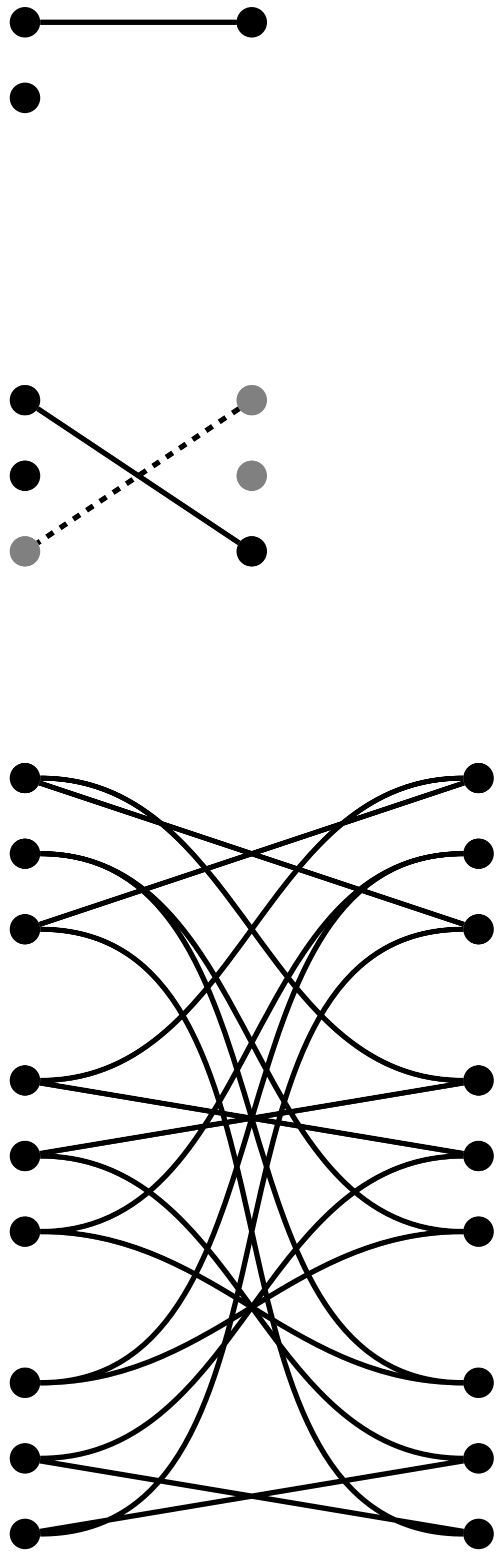

Figure 1.5: An example of a universal bipartite template. Original graph G is at the top, below that follows the modified version in which the left and right part is interchangeable. The third graph is the final result which is isomorphic to three disjoin copies of C_6. A copy of the intermediate graph is placed between each pair of the three-vertex groups in the figure.

2. Connections to Complexity Theory

Definition 23. The *Graph Isomorphism Problem* is a decision problem in which we are given two graphs G_1, G_2 on the input, and we accept the input if and only if $G_1 \cong G_2$.

The original motivation for this thesis lies in the area of complexity theory. The graph isomorphism problem is of great theoretical and practical interest. From the theoretical standpoint, this problem is one of the few that have not been ruled out as being a part of the complexity class **NP**-intermediate. That is the problem lies in the class **NP**, but there is no known polynomial algorithm for it nor is there a proof that it is a **NP**-hard problem. The best known result due to Babai gives a quasi-polynomial time complexity algorithm[1]. If it is the case that $\mathbf{P} \neq \mathbf{NP}$ then by Ladner's theorem[6] the class **NP**-intermediate is non-empty. Therefore, it is only natural that problems equally difficult as the graph isomorphism problem have been studied. This study has given rise to the complexity class **GI**.

Definition 24. **GI** is the class of decision problems for which a polynomial-time Turing reduction to the graph isomorphism problem exists.

Analogously to the definitions of **NP**-hard and **NP**-complete problems there are definitions of **GI**-hard and **GI**-complete problems.

Definition 25. A decision problem A is **GI**-*hard* if for any problem from **GI** there exists a polynomial-time Turing reduction to the problem A. If A is both **GI**-hard and lies in **GI** we say that A is **GI**-*complete*.

Many problems are known to be **GI**-complete — isomorphism problems of other structures than graphs such as finite automata or lattices, isomorphism of subclasses of graphs such as bipartite Euler graphs or chordal graphs. However, there are only a few known problems in this class that are not isomorphism problems of mathematical structures. Among these there is for example the problem of testing for the self-complementarity of a graph.[10][8][2]

A good candidate for **GI**-complete problems seems to be recognizing various properties tied to graph automorphisms. Properties describing some degree of symmetry of a graph specifically could be useful since most notions of symmetry are intrinsically linked to graph automorphisms. When it comes to graphs, there is a wide range of these properties. In this thesis, we have focused primarily on vertex-transitivity which is among the weaker of those properties.

Definition 26. The *vertex-transitivity problem* is a decision problem in which the input is a graph and we accept it if and only if it is vertex-transitive.

The *graph isomorphism problem on the class of vertex-transitive graphs* is a decision problem in which we get two vertex-transitive graphs G_1, G_2 on the input and we accept the input if and only if $G_1 \cong G_2$. If the input graphs are not vertex-transitive the result is undefined.

The *labelled graph isomorphism problem* is a decision problem in which we are given two vertex-labelled graph G_1, G_2 on the input, and we accept the input if and only if there is a label preserving isomorphism between them.

Theorem 27. *The vertex-transitivity problem lies in the complexity class* **GI**.

Proof. We provide a polynomial time Turing reduction to the labelled graph isomorphism problem. That problem is well known to be **GI**-complete[10].

Let G be the input graph. Given a vertex $u \in V(G)$ let's define a vertex-labelled graph G'_u as G with labelling $l'_u \colon V(G'_u) \to \{0, 1\}$.

$$l'_u(v) = \begin{cases} 1, & \text{if } v = u \\ 0, & \text{otherwise} \end{cases}$$

Now we iterate through all pairs of vertices $u, v \in V(G)$. We use the labelled isomorphism problem oracle to ask about G'_u and G'_v. If they are rejected we also reject.

If we do not reject after iterating through all pairs of vertices we accept.

By the construction of the labelling it is clear that isomorphisms between G'_u and G'_v correspond exactly to the automorphisms of G that map u to v. From this it follows that checking if there exists such isomorphism for every pair of vertices solves our problem. □

Note that while the previous proof uses the labelled isomorphism oracle quadratically many times we could reduce this to just a linear amount of uses. Instead of trying all pairs u, v we would fix u and check all v. This is sufficient because given vertices v and w we can create an automorphism mapping v to w by composing the inverse of the automorphism mapping u to v with the automorphism mapping u to w.

Next we shall prove that recognizing and checking isomorphism of vertex-transitive graphs are two equally hard problems.

Theorem 28. *The graph isomorphism problem on the class of vertex-transitive graphs is polynomially convertible to the vertex-transitivity problem.*

Proof. We are solving the graph isomorphism problem on the class of vertex-transitive graphs using an oracle which solves the vertex-transitivity problem. Let G_1 and G_2 be vertex-transitive graphs. If they have a different number of vertices or if only one of them is connected we reject the input. Now without loss of generality assume both are connected. If they are not, we take their complements. We construct graph H as the disjoint union of G_1 and G_2. We use the vertex-transitivity oracle on H and accept if and only if the oracle accepts its input.

First we prove that if we accept then the graphs are isomorphic. If we accept then the vertex-transitivity oracle accepted H as vertex-transitive. Therefore, for some $v_1 \in V(G_1), v_2 \in V(G_2)$ there exists an automorphism f of H which maps v_1 to v_2. Since both G_1 and G_2 are connected this means that $f[V(G_1)] = V(G_2)$. This together with the fact that $|V(G_1)| = |V(G_2)|$ and the fact that f is injective means that f restricted to G_1 is an isomorphism between G_1 and G_2.

Next we prove that if the graphs are isomorphic we accept the input. Isomorphic graphs are of the same order and are either both connected or both disconnected so the early rejection cannot happen. Thus we need to prove that under the assumption of G_1 and G_2 being isomorphic H is vertex-transitive. If we are given $v_1, v_2 \in V(G_1)$ by vertex-transitivity of G_1 there exists an automorphism f of G_1 which maps v_1 to v_2. We can extend this automorphism to H by having it be identity on vertices of G_2. Similarly, we cover the case $v_1, v_2 \in V(G_2)$. The only case left is the two vertices being in different components of H. Without loss of generality $v_1 \in V(G_1), v_2 \in V(G_2)$. Due to G_1 and G_2 being isomorphic there exists an isomorphism $g\colon G_1 \to G_2$. By vertex-transitivity of G_1 there exists an automorphism f of G_1 which maps v_1 to $g^{-1}(v_2)$. Again we extend f to H as before and extend g to H by $g(v) = g^{-1}(v)$ for $v \in V(G_2)$. Composing $g \circ f$ gives us an automorphism of H for which $(g \circ f)(v_1) = g(f(v_1)) = g(g^{-1}(v_2)) = v_2$. This concludes the proof of H being vertex-transitive so it gets accepted by the oracle.

Checking connectedness, making a graph complement and creating a disjoint union of graphs can all be done trivially in linear time. □

The graph isomorphism problem on the class of vertex-transitive graphs is not currently known to be **GI**-complete[1].[2] We have not been successful at proving that this is the case. However, we shall provide an approach which could lead to a proof of this statement if a construction of vertex-transitive supergraphs satisfying some properties is ever found.

Let us first describe the yet unfound vertex-transitive supergraph construction which would be necessary for the proof of **GI**-completeness. Let it be a mapping f which given a graph G outputs $f(G)$ a vertex-transitive supergraph of G. In order for it to be useful in the complexity theory arguments it needs to be computable in polynomial time. Moreover, in order not to require pre-existing canonical labelling (which would imply fast solution to the graph isomorphism problem) we also require that $G \simeq H \iff f(G) \simeq f(H)$ for any graphs G, H.

It is relatively easy to find an polynomial time injective mapping between general graphs and bipartite graphs, so it is sufficient to define f just for bipartite graphs and then extend this via the injective mapping. Considering the previous section this seems like a good result. However, the last requirement of f we stated is not satisfied by the bipartite construction provided in this thesis. It can be easily seen that this construction depends on the ordering $[n]$ of vertices and is invariant only under its cyclic rotations.

[1] However, some subclasses are easily recognizable. For example circulant graphs are recognizable in polynomial time.[4]

Conclusion

We have been able to provide an exponentially-sized vertex-transitive supergraph construction. For bipartite graphs we have been able to improve this result considerably by making the resulting supergraph of quadratic size. A quadratic lower bound has also been proven for general graphs. Moreover, the template schema system and properties proven about it here could be of use in possible future research into this area.

The original ambitious goal of proving that recognizing vertex-transitive graphs is **GI**-complete has not been achieved. However, we hope that groundwork laid out in this thesis brings us at least a few steps closer to this goal.

Future Work

As stated previously, the most significant avenue of future research is proving that vertex-transitive graph recognition is **GI**-complete. We believe that given a sufficiently well-behaved construction of (bipartite) vertex-transitive supergraphs, this could be proven using methods outlined in the second section.

However, the failure to provide this result in this thesis leads us to hypothesize that this construction might not be possible via the template schema approach used in this thesis. It might be worthwhile to make attempts to develop an argument proving that this particular class of supergraph construction is futile for this task. We have briefly explored group-theoretic means of proving this, but concluded that further research is out of the scope of this thesis.

Both lower and upper bounds on the order vertex-transitive supergraphs also leave room to improve. For general graphs the gap between the quadratic lower bound and the exponential upper bound is almost certainly improvable. On the other hand, we have not been able to find any non-trivial lower bound for bipartite graphs. Considering the quadratic lower bound on general graphs it is not unreasonable to expect bipartite graphs to have a similar bound which would solidify the quadratic construction as asymptotically optimal.

Finally, the methods used here could be adapted to stronger notions of graph symmetry. It is known that via a Kneser graph construction every graph has a symmetric supergraph.[5] This is once again an exponential construction and could likely be improved upon. Constructions for Cayley supergraphs, arc-transitive supergraphs or k-transitive supergraphs have not yet been attempted at all according to our research.

Bibliography

[1] László Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '16, page 684–697, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341325. doi: 10.1145/2897518.2897542. URL `https://doi.org/10.1145/2897518.2897542`.

[2] Kellog S Booth and Charles J Colbourn. *Problems polynomially equivalent to graph isomorphism*. Computer Science Department, Univ., 1979.

[3] Reinhard Diestel. *Graph Theory (Graduate Texts in Mathematics)*. Springer, August 2005. ISBN 3540261826. URL `http://www.amazon.ca/exec/obidos/redirect?tag=citeulike04-20&path=ASIN/3540261826`.

[4] S Evdokimov and Ilia Ponomarenko. Circulant graphs: Recognizing and isomorphism testing in polynomial time. *St. Petersbg. Math. J.*, 15:813–835, 01 2003. doi: 10.1090/S1061-0022-04-00833-7.

[5] P. Hamburger, A. Por, and M. Walsh. Kneser representations of graphs. *SIAM Journal on Discrete Mathematics*, 23(2):1071–1081, 2009. doi: 10.1137/080722898. URL `https://doi.org/10.1137/080722898`.

[6] Richard E. Ladner. On the structure of polynomial time reducibility. *J. ACM*, 22(1):155–171, January 1975. ISSN 0004-5411. doi: 10.1145/321864.321877. URL `https://doi.org/10.1145/321864.321877`.

[7] Robin Langer. Symmetric Graphs and their Quotients. *arXiv e-prints*, art. arXiv:1306.4798, Jun 2013.

[8] Gary L. Miller. Graph isomorphism, general remarks. *Journal of Computer and System Sciences*, 18(2):128–142, 1979. ISSN 0022-0000. doi: https://doi.org/10.1016/0022-0000(79)90043-6. URL `https://www.sciencedirect.com/science/article/pii/0022000079900436`.

[9] Gert Sabidussi. Vertex-transitive graphs. *Monatshefte für Mathematik*, 68:426–438, 1964. URL `http://eudml.org/doc/177267`.

[10] V. N. Zemlyachenko, N. M. Korneenko, and R. I. Tyshkevich. Graph isomorphism problem. *Journal of Soviet Mathematics*, 29(4):1426–1481, May 1985. ISSN 1573-8795. doi: 10.1007/BF02104746. URL `https://doi.org/10.1007/BF02104746`.

List of Figures