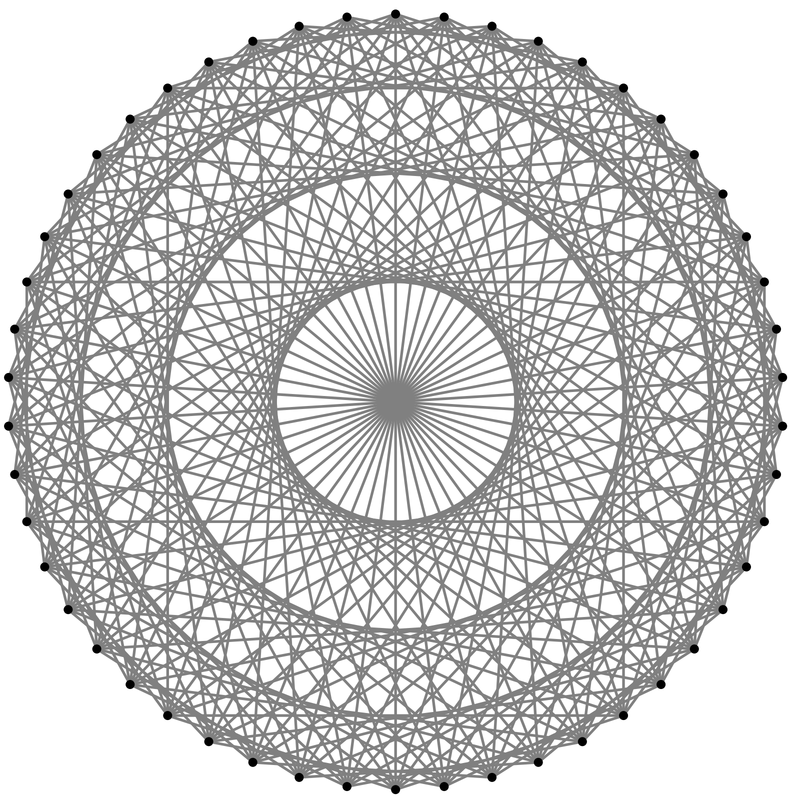
Radboud University

Word-representable graphs

Master thesis
Mathematics
Faculty of Science, Radboud University Nijmegen

September 18, 2018



Author: Bas Broere
Student number: s4137396
Supervisors: Wieb Bosma and Hans Zantema

Preface

It probably won't surprise anyone who knows me that I love all kinds of puzzles, from basic numerical puzzles up to complex problems that require some programming to figure them out. Therefore, when I was looking for a subject for my master thesis I knew exactly who to turn to, as Wieb Bosma had often supplied me with interesting puzzles during his courses. He told me about the subject of word-representable graphs, which he had heard of recently as Hans Zantema had given a presentation about this topic and his contributions, see [1], as part of a seminar at the Radboud University in Nijmegen.

Shortly thereafter I was presented with a series of questions from Hans about word-representable graphs, in the hope that they would intrigue me. One popped out almost immediately: Is the representation number of the n-cube equal to n? In my naivety, my initial reaction was that this could not be very hard to prove, and fortunately, I was wrong.

After a great start, resulting in the final subject of this thesis and an article, [4], (a collaboration of Hans Zantema and me), I had to change focus to some other problems. This led to me exploring several ways of creating representations for graphs and eventually resulted in a general way of representing bipartite graphs.

It would be an understatement when I say that I enjoyed doing this research, as I truly wish that I could spend more time on it. Brainstorming with Hans and Wieb led to a constant stream of new ideas, possibilities and generalisations, and they were a great inspiration to me during the process. Their positive attitude and contagious curiosity helped me push the boundaries of my capabilities and stay curious myself.

I would also like to take this opportunity to thank Lieke-Rosa Koetsier and Paulien Schets, who were there to help me find out why something did or did not work, often accompanied by a nice cup of tea or hot chocolate.

Lastly I would like to thank Sergey Kitaev and William Trotter for providing me with the information I needed to expand my research.

I hope you enjoy reading,
Bas Broere

Contents

1 Introduction **3**

2 Graph theory **4**
2.1 Terminology . . . 4
2.2 Operations and inclusions . . . 6
2.3 Directed graphs . . . 8

3 Word-representations **12**
3.1 Terminology . . . 12
3.2 Basic results . . . 14
3.3 Occurrence-based functions . . . 15

4 Constructing representations **18**
4.1 Induced subgraphs and minors . . . 18
4.2 Non-word-representable graphs . . . 20
4.3 Generic constructions . . . 21
4.3.1 Transitive orientations . . . 21
4.3.2 Semi-transitive orientations . . . 24
4.4 Bipartite graphs . . . 30
4.5 Existing constructions . . . 32
4.5.1 Trees . . . 33
4.5.2 Cycles . . . 34
4.5.3 Ladders . . . 35
4.5.4 Prisms . . . 36

5 Cartesian product **37**
5.1 Cartesian product with K_2 . . . 37
5.1.1 Representation . . . 37
5.1.2 Cubes and prisms . . . 39
5.2 Cartesian product with K_n . . . 41
5.3 Cartesian product with a permutationally representable graph . . . 43

6 Open problems **47**
6.1 Bipartite graphs . . . 47
6.2 Cartesian products . . . 47

1 Introduction

Suppose we have a graph $G = (V, E)$. A word w over the alphabet V is said to represent G if and only if for every edge $\{x, y\} \in E$ the letters x, y alternate in w. This means that when we only look at the letters x and y in w, we find the pattern $xyxy \ldots$ or $yxyx \ldots$ of even or odd length. The graph G is called word-representable if and only if such a word exists.

A lot of research has been done on the subject of word-representable graphs and ways to construct their representants. This thesis lists some of this research and some new results in representing graphs, and is structured as follows. Sections 2 and 3 focus on preliminaries in the fields of graph theory and word-representable graphs, where we will also introduce the new concept of occurrence-based functions. This will be followed by Chapter 4, in which we will take a closer look at existing constructions using the new terminology. Furthermore we will discuss a new result on representing bipartite graphs. The focus of Chapter 5 is the representation of the Cartesian product of two graphs. We will finish by stating some open problems that result from this research.

Many of the results of Chapter 5 also appeared in [4], but we will make a further generalisation. Note that because of this, many definitions, lemmas and theorems in this thesis are formulated in the same way as in [4].

2 Graph theory

In this chapter we will discuss basic definitions and theorems in the field of graph theory. Most of these definitions and results originate from [2]. We will give examples where necessary.

We start with the basic definition of a graph, after which we will talk about operations and inclusions. Lastly we will discuss directed graphs.

2.1 Terminology

Definition 2.1. A *graph* $G = (V, E)$ is defined by the two sets V of *nodes* and E of *edges*, where E consists of unordered pairs of different nodes.

Remark 2.2. This definition does not allow multiple edges between two nodes or an edge from a node to itself.

Example 2.3. Let $G = (\{1, 2, 3, 4, 5\}, \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}\})$ be a graph. A graph can be graphically represented in an intuitive way. We do this by drawing points for nodes, labelling them with the symbols in V, and drawing a line for all edges between two points x and y if and only if $\{x, y\} \in E$. Figure 1 represents G.

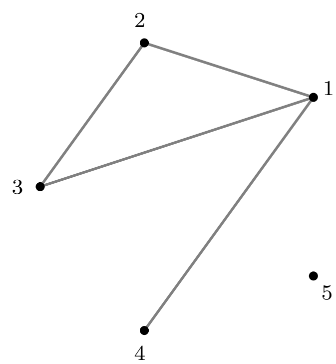


Figure 1: Graphical representation of
$G = (\{1, 2, 3, 4, 5\}, \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}\})$.

We see that different nodes may have different numbers of edges connected to it.

Definition 2.4. The *neighbourhood* of a node x in the graph $G = (V, E)$ is the set $N(x) = \{y \in V \mid \{x, y\} \in E\}$. The *degree* of a node is the number of elements, $\#N(x)$, of its neighbourhood. For a given $x \in V$ all edges in the set $\{\{x, y\} \mid y \in N(x)\}$ are *incident to* x.

Node 5 in the graph of Example 2.3 has degree zero, node 4 has degree one, nodes 2 and 3 have degree two, node 1 has degree three and $N(1) = \{2, 3, 4\}$.

There are a lot of different families of graphs. One example, which we will see multiple times throughout this thesis, is the family of complete graphs.

Definition 2.5. A graph $G = (V, E)$ is a *complete graph* if for all $x, y \in V$ it holds that $\{x, y\} \in E$. A complete graph is denoted by K_n, where $n = \#V$.

Another example is formed by the cycle graphs, which can be drawn in a circular fashion.

Definition 2.6. A graph $G = (V, E)$ is a *cycle graph* if $V = \{x_1, x_2, \dots, x_n\}$ for some n, and $E = \{\{x_1, x_2\}, \{x_2, x_3\}, \dots, \{x_{n-1}, x_n\}, \{x_1, x_n\}\}$.

Some families of graphs allow us to say something more generic about properties a large collection of graphs can have. For example, for certain families of graphs we can say something about the chromatic number of the graphs in that family.

Definition 2.7. A graph $G = (V, E)$ is *k-colourable* if and only if every node can be given one out of k colours such that x and y have a different colour if $\{x, y\} \in E$. The smallest k such that a graph G is k-colourable is called the *chromatic number* of the graph.

The decision problem if for a $k \geq 3$, a graph is k-colourable is an NP-complete problem [8]. However, we are able to say something about the chromatic number of graphs that belong to certain families of graphs. For instance, *planar graphs*, graphs that can be drawn in $\mathbb{R}^2$ without any two edges crossing, are all 4-colourable [17]. Another example of this is the family of bipartite graphs, all members of which are 2-colourable in an obvious way.

Definition 2.8. A graph $G = (V, E)$ is a *bipartite graph* if the set of nodes can be split into two disjoint subsets, the *parts*, of nodes $V = A \cup B$ such that there are no edges between nodes in A and no edges between nodes in B.

A bipartite graph $G = (V, E)$ is a complete bipartite graph if $\{x, y\} \in E$ for all $x \in A$ and $y \in B$. A bipartite graph with $\#A = n$ and $\#B = m$ is denoted by $K_{n,m}$.

Example 2.9. The graph in Figure 2 is bipartite. This graph is 2-colourable, as shown in Figure 2 with the colours red and blue.

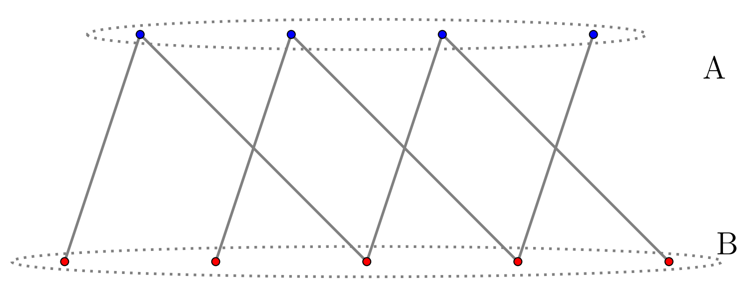


Figure 2: A bipartite graph.

Remark 2.10. The family of bipartite graphs is not only a subset of the 2-colourable graphs, they are in fact equal.

Lastly we will look at walks and paths.

Definition 2.11. A *walk* in a graph $G = (V, E)$ is a series of nodes $x_1, x_2, \dots, x_n$ such that $\{x_i, x_{i+1}\} \in E$ for all $1 \leq i < n$.

Definition 2.12. A *path* between two different nodes x and y in a graph G is a walk $x_1, x_2, \dots, x_n$ such that $x_1 = x$, $x_n = y$ and for all i, j $x_i \neq x_j$.

Definition 2.13. A graph $G = (V, E)$ is *connected* if there is a path between every pair of nodes $x, y \in V$.

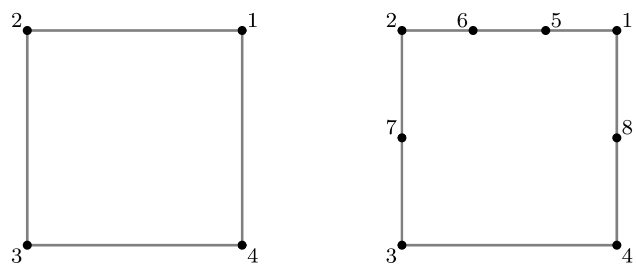
Example 2.14. The nodes $4, 1, 2, 1, 3$ form a walk in the graph of Figure 1. This is not a path, as node 1 occurs multiple times. The nodes $3, 2, 1, 4$ do form a path. This graph is not connected, as there is no path between node 5 and any other node.

2.2 Operations and inclusions

There are many ways to obtain a graph from other graphs. For instance, we can add or remove edges and nodes, see Section 4.1. A lesser known operation is constructing a subdivision of a graph.

Definition 2.15. A graph $H = (V_H, E_H)$ is a *subdivision* of a graph $G = (V_G, E_G)$ if $V_G \subseteq V_H$ and $\{x, y\} \in E_G$ if and only if there is a set $\{x_1, x_2, \ldots, x_n\} \subseteq V_H$ such that $x_1 = x$, $x_n = y$, $\{x_i, x_{i+1}\} \in E_H$ for all $1 \leq i < n$ and $x_i \notin V_G$ for all $1 < i < n$.

Example 2.16. Figure 3 shows a graph and a subdivision of that graph. We see that we can look at a subdivision of a graph as adding extra nodes on already existing edges, effectively splitting them in multiple pieces.



(a) Original graph. (b) Subdivision.

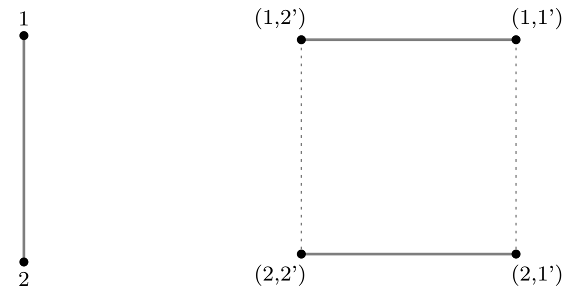
Figure 3: Example of a subdivision.

Remark 2.17. Example 2.16 shows that a subdivision of a graph has the same general structure as the original graph. There are also certain properties that are preserved when constructing a subdivision, for example it is trivial to see that a subdivision of a planar graph is again planar.

Another graph operation is taking the Cartesian product of two graphs, which will be the main subject of Chapter 5.

Definition 2.18. The *Cartesian product* of two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$ is defined as $G \,\square\, H = (V_{G \square H}, E_{G \square H})$, where $V_{G \square H} = V_G \times V_H$ and
$E_{G \square H} = \{\, \{\, (x, x')\,,\, (y, y')\,\} \mid x = y \text{ and } \{x', y'\} \in E_H, \text{ or } x' = y' \text{ and } \{x, y\} \in E_G\}$.

Example 2.19. We are going to construct the Cartesian product of $G = (\{1, 2\}, \{\{1, 2\}\})$, with itself, see Figure 4a. To make matters more clear, we will denote the second graph as $G' = (\{1', 2'\}, \{\{1', 2'\}\})$. Using the definition we find $V_{G \square G'} = \{(1, 1'), (1, 2'), (2, 1'), (2, 2')\}$ and $E_{G \square G'} = \{\{(1, 1'), (1, 2')\}, \{(2, 1'), (2, 2')\}, \{(1, 1'), (2, 1')\}, \{(1, 2'), (2, 2')\}\}$, see Figure 4b.



(a) $G = (\{1,2\}, \{\{1,2\}\})$. (b) $G \,\square\, G'$.

Figure 4: Example of the Cartesian product of two graphs.

Remark 2.20. Often we do not name the nodes (x, y) explicitly, but just assign numbers to them.

The Cartesian product of a graph is interesting because it is easy to find copies of the original graphs in the product. The nodes in the Cartesian product have names (x, y) where x is a node in G and y is a node in H. When we restrict the graph to all nodes (x, y) for a fixed y, we find a copy of G and when we do the same for a fixed x, we find a copy of H. This is illustrated in Figure 5.

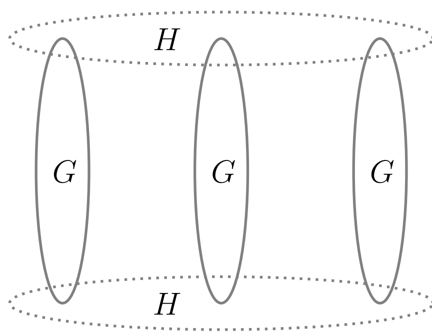


Figure 5: Cartesian product of two graphs, G and H.

We see that the graphs G and H are in a way included in, or are a subgraph of, $G \,\square\, H$.

Definition 2.21. A graph $H = (V_H, E_H)$ is a *subgraph* of a graph $G = (V_G, E_G)$ if $V_H \subseteq V_G$ and $E_H \subseteq E_G$.

This means that a subgraph is obtained by removing zero or more edges and zero or more nodes from G. The only thing we need to think about is that when we remove a node, we also remove all the edges indicent to that node. When we restrict ourselves to exclusively remove nodes and their incident edges, we come to the following definition.

Definition 2.22. A subgraph $H = (V_H, E_H)$ of $G = (V_G, E_G)$ is an *induced subgraph* if for all $x, y \in V_H$ it holds that $\{x, y\} \in E_H$ if and only if $\{x, y\} \in E_G$.

We see that the graphs G and H are induced subgraphs of $G \,\square\, H$. We illustrate these definitions further in Example 2.27.

Definition 2.23. If a family of graphs is closed under taking induced subgraphs, the family is called a *hereditary family.*

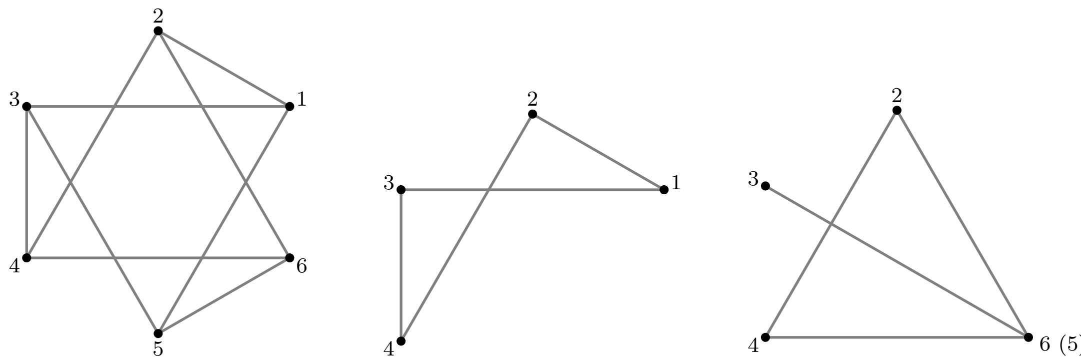
Example 2.24. The families of planar graphs and bipartite graphs are examples of hereditary families of graphs.

The following gives a more complex form of the inclusion of a graph in another graph.

Definition 2.25. An *edge contraction* of an edge $\{x, y\}$ in a graph $G = (V, E)$ results in the graph $G' = (V', E')$ where $V' = V \setminus \{y\}$ and $E' = \{\{x, x'\} \mid x' \in N(y)\} \cup E \setminus \{\{x, y\}\}$.

Definition 2.26. A *minor* of the graph G is any graph H that can be obtained from G by contracting edges, removing edges and removing nodes and their incident edges.

Example 2.27. Figure 6 shows a graph, an induced subgraph and a minor of that graph.



(a) Original graph.

(b) An induced subgraph with nodes 1, 2, 3 and 4.

(c) A minor obtained by removing node 1, the edge $\{3, 4\}$ and contracting the edge $\{5, 6\}$.

Figure 6: Example of an induced subgraph and a minor.

The inclusion of one graph in another graph can tell us something about certain properties of (a family of) graphs. The following theorems are examples of this.

Theorem 2.28. (Wagner's theorem, [2]) A finite graph is a planar graph if and only if it does not contain the graphs K_5 or $K_{3,3}$ as a minor.

Theorem 2.29. (Kuratowski's theorem, [2]) A finite graph is a planar graph if and only if it does not contain a subdivision of K_5 or $K_{3,3}$ as subgraph.

2.3 Directed graphs

We start by stating the definition of a directed graph and some basic properties.

Definition 2.30. A *directed graph* $D = (W, A)$ is defined by the two sets W of *nodes* and A of *arrows*, where A consists of ordered pairs of different nodes (x, y), indicating there is an arrow from x to y, which is often denoted as $x \to y$.

Remark 2.31. This definition does allow for multiple arrows between two nodes. Even though this is allowed, we will focus on directed graphs where only a single or no arrow between two nodes is allowed.

Analogous to undirected graphs, we can also speak about walks and paths in directed graphs.

Definition 2.32. A *directed walk* in a directed graph $D = (W, A)$ is a series of nodes $x_1, x_2, \ldots, x_n$ such that $(x_i, x_{i+1}) \in A$ for all $1 \leq i < n$.

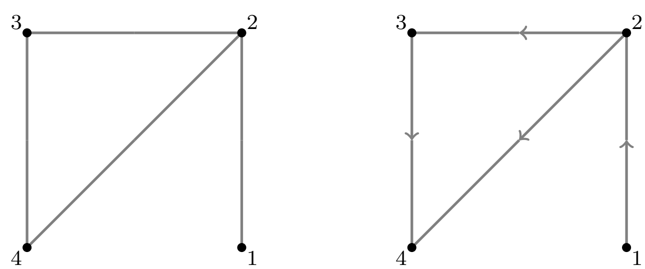
Definition 2.33. A *directed path* between two different nodes x and y in a directed graph $D = (W, A)$ is a directed walk $x_1, x_2, \ldots, x_n$ such that $x_1 = x$, $x_n = y$ and for all i, j $x_i \neq x_j$. When there is a directed path from a node x to y, this is often denoted by $x \rightsquigarrow y$.

Remark 2.34. By defining directed paths this way, we do not allow for paths between a node and itself. This becomes important in the next definition.

Definition 2.35. A directed graph $D = (W, A)$ is *acyclic* if for all $x, y \in W$, when there is a directed path from x to y, there is *no* directed path from y to x.

Remark 2.36. It is easy to see that if a directed graph is acyclic, then every walk between two different nodes is a path.

Example 2.37. Figure 7b shows a directed graph. We see that there are two directed paths from 2 to 4 (2,3,4 and 2,4), while there is no directed path from 2 to 1.



(a) An undirected graph. (b) A directed graph.

Figure 7: Example a graph and a directed graph.

Directed graphs are a useful way of visualising processes. The graph in Figure 7b can be interpreted as the path a product has to go through in a factory. Every product starts at unpacking (1) and proceeds to sorting (2). If the product needs cleaning it has to go to (3) and after that it can go to distribution (4). This way of looking at a directed graphs adds meaning to the direction of the arrows of the graph.

We will now look at a way to turn an undirected graph into a directed graph. This is done by assigning a direction to every edge in an undirected graph, essentially turning every edge in an arrow.

Definition 2.38. An *orientation* $\mathcal{O}$ of an undirected graph $G = (V, E)$ is a map that results in a directed graph $G_{\mathcal{O}}$ by mapping each edge $\{x, y\}$ to either (x, y) or (y, x). We call G the *underlying undirected graph* of $G_{\mathcal{O}}$.

Example 2.39. An example of an orientation of the graph in Figure 7a is the graph in Figure 7b.

There are two kinds of orientations we will take a closer look at.

Definition 2.40. ([7]) A directed graph $D = (W, A)$ is called *transitive* if the following property holds: If $(x, y) \in A$ and $(y, z) \in A$, then $(x, z) \in A$. An orientation $\mathcal{O}$ of a graph G is *transitive* if $G_{\mathcal{O}}$ is transitive.

Definition 2.41. ([7]) A directed graph $D = (W, A)$ is called *semi-transitive* if the following properties hold:

- D acyclic;
- For every directed path $x_1, x_2, \ldots x_n$ in D one of the following two possibilities holds:
  - $(x_1, x_n) \notin A$;
  - $(x_1, x_n) \in A$ and $(x_i, x_j) \in A$ for all $1 \leq i < j \leq n$.

An orientation $\mathcal{O}$ of a graph G is *semi-transitive* if $G_{\mathcal{O}}$ is semi-transitive.

It is easy to see that every transitive orientation is semi-transitive. The following example illustrates that the converse is not true.

Example 2.42. The orientation $\mathcal{O}$ of the graph G shown in Figure 7b is not transitive, as $(1, 2), (2, 3) \in A$, but $(1, 3) \notin A$, where $G_{\mathcal{O}} = (W, A)$. The orientation is, however, semi-transitive, which we will now show.

It is easy to see that $G_{\mathcal{O}}$ is acyclic. For paths consisting of only two nodes the second property holds automatically since $G_{\mathcal{O}}$ is acyclic. We will now look at all the directed paths that consist of at least three nodes:

- (1,2,3,4): $(1, 4) \notin A$ so the property holds;
- (1,2,4): Again $(1, 4) \notin A$, so the property holds;
- (2,3,4): $(2, 4) \in A$, so the property holds.

So we conclude that the orientation in Figure 7b is semi-transitive.

Remark 2.43. We say that a graph G *admits a (semi-)transitive orientation* if there exists a (semi-)transitive orientation $\mathcal{O}$ of G.

Example 2.44. As we have seen in Example 2.42, the graph in Figure 7a admits a semi-transitive orientation. The orientation shown in Figure 7b is not transitive, but the graph does admit a transitive orientation, see Figure 8.

There are families of graphs of which it is known that every member admits a transitive orientation, for example the bipartite graphs. The same can be said for semi-transitive orientations, but this set of graphs is much larger.

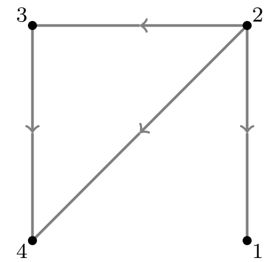


Figure 8: A transitive orientation of the graph in Figure 7a.

3 Word-representations

In this chapter we will talk about words and graphs. We will focus mainly on terminology and basic results, but we will also introduce the new concept of occurrence-based functions. Most of the definitions, lemmas and theorems in Sections 3.1 and 3.2 are based on [12].

As a convention, symbols like x and y will denote letters, while symbols like w and v denote words. Furthermore let ϵ be the empty word. When we write two or more words or letters after each other, like wv or xxy, we mean the concatenation of these letters or words. When we speak about an alphabet, we mean a finite set of symbols, for example $\{a, b, c, d, e\}$ or $\{1, 2, 3, 4\}$.

3.1 Terminology

We start with some definitions regarding words.

Definition 3.1. For a word w over an alphabet A, two letters x and y are said to *alternate* in w if between every two x's in w a y occurs and between every two y's in w an x occurs.

Stated otherwise: removing all letters but x and y from w results in a word $xyxy\dots$ or $yxyx\dots$ of even or odd length.

Definition 3.2. A word w over an alphabet A is called *k-uniform* if every $x \in A$ occurs exactly k times in w. A 1-uniform word over A is called a *permutation of* A.

Definition 3.3. If w is a word over an alphabet A, and $B \subseteq A$, then the word w_B is defined as the word obtained by removing all letters in $A \setminus B$ from w.

Remark 3.4. Two letters x and y alternate in a k-uniform word w if and only if $w_{\{x,y\}}$ is either $(xy)^k$ or $(yx)^k$.

Definition 3.5. The *initial permutation*, $p(w)$, of a word w is obtained by removing all but the first occurrence of each letter in w.

Definition 3.6. A word v is a *rotation* of the word $w = x_1x_2\dots x_n$ if there is an i such that $v = x_ix_{i+1}\dots x_nx_1\dots x_{i-1}$.

Example 3.7. In the word $w = 12341432$ the letters 1 and 2 alternate, as $w_{\{1,2\}} = 1212$, but the letters 3 and 4 do not, as $w_{\{3,4\}} = 3443$. The word w is 2-uniform as every letter occurs two times. Also, the initial permutation of this word is $p(w) = 1234$ and $v = 35253212$ is a rotation of w.

We want to use a word over the alphabet V to represent a graph $G = (V, E)$. We will only talk about representing undirected graphs and we will use directed graphs in the process of representing undirected graphs.

Definition 3.8. A graph $G = (V, E)$ is *word-representable* if there is a word w over the alphabet V such that:

- $\{x, y\} \in E$ if and only if x and y alternate in w;
- For all $x \in V : w_{\{x\}} \neq \epsilon$.

The word w is said to *represent*, or be a *representant* of G, and the graph that is represented by a word w is denoted by $G(w)$.

Remark 3.9. A word represents a unique graph, while a graph can have multiple words representing it. Also, a graph need not be word-representable.

Example 3.10. The graph in Figure 9a has the word $w = 431423124132$ as a representant. The graph in Figure 9b is the graph that is represented by $w = 1342132412$.

(a) Graph represented by the word $w = 431423124132$.

(b) $G(w)$ where $w = 1342132412$.

Figure 9: Examples of graph representations.

When a word needs to be manipulated in some way, it is often easier to have uniform words to work with.

Definition 3.11. A graph G is called *k-representable* if there is a k-uniform word w that represents G. The smallest k such that G is k-representable is called the *representation number* of G. By definition, non-word-representable graphs have representation number ∞.

Example 3.12. The complete graphs are the only 1-representable graphs. For every n the graph K_n can be represented by a permutation of the set $\{1, 2, \ldots, n\}$, and in particular by the word $w = 123 \ldots n$. See Figure 10 for an illustration of K_7.

As we will see in Section 4.3.1, certain graphs are word-representable as a concatenation of permutations. First we will define this formally and look at an example.

Definition 3.13. A graph $G = (V, E)$ is *permutationally representable* if there is a word $w = p_1 p_2 \ldots p_k$ such that w represents G and for every i, the word p_i is a permutation of V. The word w is called a *permutation-representant*, or *k-permutation-representant*. The smallest number of permutations needed to represent G is called the *permutation-representation number*.

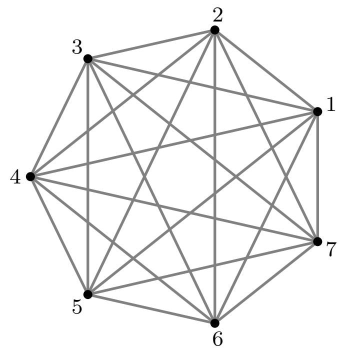


Figure 10: K_7, represented by $w = 1234567$.

Remark 3.14. A permutation-representant is always a uniform word.

Example 3.15. The graph in Figure 9a is also represented by the word $w' = 3142\mathbf{1324}$, which is a concatenation of the two permutations 3142 and 1324, so it is 2-permutation-representable.

3.2 Basic results

Now we discuss some basic results regarding word-representable graphs.

The first thing we note is that if w is a word that represents a graph G, then also the reverse of w represents G. If w is k-uniform for some k we can say even more.

Theorem 3.16. Let w be a k-uniform word that represents a graph G. Then any rotation of w represents G.

Proof. We need to prove that x and y alternate in w if and only if they alternate in every rotated version of w. It suffices to prove this for a rotation over one position to the left, i.e. $x_1x_2 \dots x_t \to x_2x_3 \dots x_tx_1$. Denote this rotated version by w'.

Assume x and y alternate in w, so without loss of generality $w_{\{x,y\}} = (xy)^k$. There are two possibilities; either w starts with x or not.

If w starts with x, then w' ends in x, and thus $w'_{\{x,y\}} = y(xy)^{k-1}x$, so x and y alternate in w'.

If w does not start with x, then $w'_{\{x,y\}} = w_{\{x,y\}} = (xy)^k$, so x and y alternate in w'.

Similarly, we see that if they alternate in w', they alternate in w. □

As noted before, we will mostly talk about uniform words for graphs, because they are easy to work with. The following lemma is an important result that we need to make sure we can always speak of uniform representants.

Lemma 3.17. ([12]) Let w be a non-uniform word representing G. Then there exists a uniform word v that represents G.

In Section 3.3 we will introduce notation that makes it easier to prove this, which we will do later on. The following theorems are a direct consequence of this lemma.

Theorem 3.18. A graph G is representable if and only if it is k-representable for some $k \geq 1$.

Theorem 3.19. Every k-representable graph is also $(k+1)$-representable.

3.3 Occurrence-based functions

The focus of this section is to define an intuitive way of describing operations on words. We will use the notion in upcoming chapters to reformulate some already existing constructions of words for graphs and devise new constructions.

The way that we will do this is by defining occurrence-based functions. The notion of occurrence-based functions was developed together with Hans Zantema in order to formalise notations in [4].

Definition 3.20. Let V and V' be (possibly different) alphabets, and let $N_k = \{1, \ldots, k\}$. The *labelling function* of finite words over V is defined by $H : V^* \to (V \times N_k)^*$, where the ith occurrence of each letter x is mapped to the pair (x, i), and k satisfies the property that every symbol occurs at most k times in w. Now $H(w)$ is called the *labelled version of* w. An occurrence-based function is the composition $(h \circ H)$ of a homomorphism $h : (V \times N_k)^* \to (V')^*$ and the labelling function H. As a shorthand we will write $h(w)$ instead of $h(H(w))$.

Example 3.21. Recall from Definition 3.5 that the initial permutation of a word w was defined as removing all but the first occurrence of every letter from w. This is equivalent to applying the following occurrence-based function:

$$h(x, i) = \begin{cases} x & \text{if } i = 1 \\ \epsilon & \text{otherwise.} \end{cases}$$

So $h(512356324215621) = 512364 = p(512356324215621)$.

The occurrence-based function used in Example 3.21 gives rise to the following generalisation of the initial permutation.

Definition 3.22. For a k-uniform word w and a set $A \subseteq N_k = \{1, \ldots, k\}$, the occurrence based function p_A is defined, for every symbol x, by $p_A(x, i) = x$ for all $i \in A$, and $p_A(x, i) = \epsilon$ for all $i \notin A$. In the case that $A = \{i\}$, for some i, we call $p_{\{i\}} = p_i$ the *ith permutation* of w.

It allows for the following useful lemma by Hans Zantema, a generalisation of previously developed lemmas.

Lemma 3.23. ([4]) Let w be a k-uniform word representing a graph G. For some $m > 1$ let $A_1, \ldots, A_m$ be non-empty subsets of $N_k = \{1, \ldots, k\}$ such that for all $j = 1, \ldots, k-1$ there exists an $i \in \{1, \ldots, m\}$ for which $\{j, j+1\} \subseteq A_i$. Then the $(\sum_{i=1}^{m} \#A_i)$-uniform word $w' = p_{A_1}(w) p_{A_2}(w) \cdots p_{A_m}(w)$ also represents the graph G.

Proof. We prove that any two symbols x and y alternate in w if and only if they alternate in w'.

First assume they alternate in w, then $w_{\{x,y\}}$ is either $(xy)^k$ or $(yx)^k$. Without loss of generality assume $w_{\{x,y\}}(xy)^k$. Then $p_{A_i}(w)_{\{x,y\}} = (xy)^{\#A_i}$ for all $i = 1, \ldots, m$, so $w'_{\{x,y\}} = (xy)^{\sum_{i=1}^{m} \#A_i}$, by which x and y alternate in w'.

Conversely, assume x and y alternate in w'. Then either $p_{A_i}(w)_{\{x,y\}} = (xy)^{\#A_i}$ for all $i = 1, \ldots, m$, or $p_{A_i}(w)_{\{x,y\}} = (yx)^{\#A_i}$ for all $i = 1, \ldots, m$. Without loss of generality assume the first. Let $\{j, j+1\} \subseteq A_i$, then from $p_{A_i}(w)_{\{x,y\}} = (xy)^{\#A_i}$ we conclude that

- the jth x in w is left from the jth y in w;
- the jth y in w is left from the $(j+1)$th x in w;
- the $(j+1)$th x in w is left from the $(j+1)$th y in w.

As it is assumed for all $j = 1, \ldots, k-1$ there is such an A_i, we obtain this property for all $j = 1, \ldots, k-1$, from which we conclude that x and y alternate in w. □

Corollary 3.24. Let w be a k-uniform word representing a graph G. The word $p_A(w)w$ also represents G for all $A \subseteq N_k$.

From now on, wherever possible, we will use occurrence-based functions in constructions. We will start by proving Lemma 3.17.

Lemma. ([12]) Let w be a non-uniform word representing G. Then there exists a uniform word v that represents G.

Proof. We will construct the word v as follows:

1. Initialise w' as w;
2. Find the maximum integer k such that there is a letter x in w' that occurs k times;
3. Define B as the set of all letters that occur fewer than k times in w';
4. Replace w' by $p_1(w')_B w'$;
5. If w' is uniform, $v = w'$, otherwise go back to step 3.

As the word we consider is finite and it is clear that we add only letters that occur fewer than k times, this construction terminates and thus the resulting word is k-uniform where k is the number found in step 2.

We need to prove that v represents G, so x and y alternate in v if and only they alternate in w.

Assume x and y alternate in v. As w' is initialised as w we observe that w' has w as suffix. This directly implies that x and y must alternate in w.

Now assume x and y alternate in w and they occur an equal number of times in w. If $w_{\{x,y\}} = (xy)^t$ for some t, then $p(w)_{\{x,y\}} = xy$ and in every iteration of the construction $p(w')_{\{x,y\}} = xy$ or $p(w')_{\{x,y\}} = \epsilon$. From this we conclude that $w'_{\{x,y\}} = (xy)^s$ for some s and thus x and y alternate in v.

Now assume x and y alternate in w and, without loss of generality, x occurs more often than y. Note that if x occurs t times, then y occurs $t-1$ times and $w_{\{x,y\}} = (xy)^t x$ by the assumption that x, y alternate. The letter y must be added to once more than x. In the first iterations we again see that $p(w')_{\{x,y\}} = xy$. After a certain point, x occurs k times, but now y occurs $k-1$ times, so this results in $p(w')_B = y$, followed by making the final word $v_{\{x,y\}} = (yx)^k$. □

Remark 3.25. Using Lemma 3.17 we find a word v that is k-uniform where k is the maximum found in step 2 in the proof. This is not necessarily the smallest uniform possible, as there are no requirements for w to be small.

4 Constructing representations

In this chapter we will take a look at different constructions of words for graphs. We will start by discussing induced subgraphs and minors, as this will help in determining whether or not a graph is word-representable. After this, we discuss some non-word-representable graphs and take a look at general ways to represent graphs that are word-representable, including a new result on representing bipartite graphs. Lastly, we will look at some existing constructions for words-representations and reformulate them using occurrence-based functions.

4.1 Induced subgraphs and minors

In Section 2.2 we defined (induced) subgraphs and minors of a graph. One of the first questions we might ask ourselves is if there is an equivalent to Theorem 2.28 or 2.29 for word-representable graphs. To find the answer to this question we first need to look at the involved operations.

It is easy to see that if we remove a node x, together with all its incident edges, from a graph G that is word-representable, we obtain a new word-representable graph.

Theorem 4.1. Let $G = (V_G, E_G)$ be a graph represented by the word w and let $H = (V_H, E_H)$ be an induced subgraph of G. Then $w' = w_{V_H}$ represents H.

Proof. The word w' only consists of letters in V_H by definition.

Now we need to prove that $\{x, y\} \in E_H$ if and only if x and y alternate in w'. Let $x, y \in V_H$. As H is an induced subgraph of G, we know that $E_H \subseteq E_G$, so $\{x, y\} \in E_H$ if and only if $\{x, y\} \in E_G$. Note that $w_{\{x,y\}} = w'_{\{x,y\}}$, so x and y alternate in w if and only if they alternate in w'. □

This theorem leads to the following.

Corollary 4.2. ([12]) The family of word-representable graphs is hereditary; thus, if G is a graph and H is a non-word-representable induced subgraph of G, then G is not word-representable.

Remark 4.3. Theorem 4.1 requires H to be an *induced* subgraph. If this theorem were true for general subgraphs, then every graph would be word-representable, because every complete graph is word-representable and every graph is a subgraph of a complete graph.

There is no similar result of Theorem 4.1 for removing a single edge, so it is worthwhile to take a look at cases where removing an edge keeps the graph representable. For instance, it is easy to see that when we remove or add an edge in a bipartite graph, we obtain another bipartite graph, which is word-representable, see Section 4.4.

Theorem 4.4. Let $G = (V, E)$ be a graph, $\{x, y\} \in E$ and w a representant of G that contains x and y at least twice. If w has one of the following forms, then $G' = (V, E \backslash \{\{x, y\}\})$ can be represented by w':

- $w = w_1xyw_2$, then $w' = w_1yxw_2$;
- $w = xvy$, then $w' = yvx$.

The proof of this theorem is very basic and is omitted here.

Remark 4.5. The cases mentioned in Theorem 4.4 are relatively easy, but are also the most general cases known at this point in time.

Remark 4.6. In the same way as in Theorem 4.4, we can state something about adding an edge to a graph, but this theorem would have a more elaborate case distinction.

It follows from Remark 4.3 that there are cases where removing or adding an edge does not keep the graph representable. From this follows that there does not exist an equivalent of Theorem 2.29 or 2.28 for word-representable graphs.

There is one operation in the construction of a minor we have not discusses yet, and that is performing an edge-contraction. It could be that the ability to remove single edges is the only thing that prevents us from stating an equivalent to Theorem 2.28. Unfortunately, performing an edge-contraction also does not preserve word-representability in general, as we will illustrate now.

We will show this with the use of k-subdivisions, which are subdivisions in which every edge is split into at least k parts by adding at least $k - 1$ nodes on each edge.

Definition 4.7. ([13]) A graph $H = (V_H, E_H)$ is a *k-subdivision* of a graph $G = (V_G, E_G)$ if $V_G \subseteq V_H$, and $\{x, y\} \in E_G$ if and only if there is a set $\{x_1, x_2, \ldots, x_n\} \subseteq E_H$ such that $x_1 = x$, $x_n = y$, $\{x_i, x_{i+1}\} \in E_H$ for all $1 \leq i < n$, $k + 2 \leq n$ and $x_i \notin V_G$ for all $1 < i < n$.

Remark 4.8. It is easy to see that a k-subdivision of a graph is also an l-subdivision for all $l < k$.

Example 4.9. Figure 11 shows a graph and a 2-subdivision of that graph. Note that the edge $(2, 3)$ is split into three pieces, but the edge $(1, 2)$ is only split in two pieces.

(a) A graph G. (b) A 2-subdivision of G.

Figure 11: A graph G and a 2-subdivision.

The following theorem does not state anything about edge-contractions, but a direct corollary of it does.

Theorem 4.10. ([13]) For every graph G a 3-subdivision of G is 3-word-representable.

The proof of this theorem, as formulated in [13], is omitted here.

It is an immediate consequence of this theorem that applying an edge-contraction in a word-representable graph can either give a word-representable or a non-word-representable graph.

The idea behind this is the following. Assume we have a 4-subdivision H of a non-word-representable graph G. Contracting edges in H to obtain a 3-subdivision (that is not a 4-subdivision) of G results in a word-representable graph, as it still is a 3-subdivision of G. When we continue contracting edges to obtain G again, we get a non-word-representable graph.

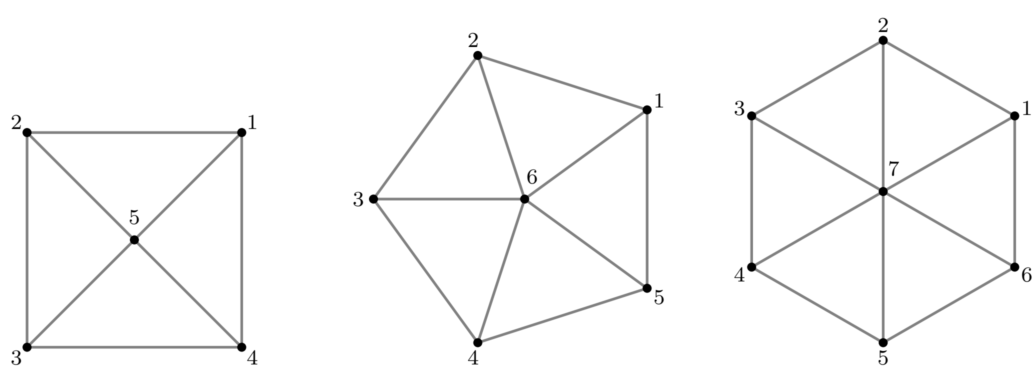
Remark 4.11. Theorem 4.10 is a very powerful theorem, because when we want to make a word-representation for a non-word-representable graph we can agree on representing a 3-subdivision of the graph instead. This will, in most cases, result in a large word. This is because every edge in the original graph gives us two more nodes and it usually holds that the more nodes there are, the larger the word will be.

4.2 Non-word-representable graphs

As stated before, not every graph is word-representable, but no non-word-representable graphs have been discussed yet. We will do this now, starting with the smallest (by number of nodes) non-word-representable graph.

Definition 4.12. ([2]) The *wheel-graph* with $n+1$ nodes, or the n-wheel, W_n is the graph with $V = \{x_1, x_2, \dots, x_n, y\}$ and $E = \{\{x_1, x_2\}, \{x_2, x_3\}, \dots, \{x_{n-1}, x_n\}, \{x_1, x_n\}\} \cup \{\{x_i, y\} \mid 1 \le i \le n\}$.

Example 4.13. In Figure 12 the 4-, 5- and 6-wheels are shown.



(a) The 4-wheel. (b) The 5-wheel. (c) The 6-wheel.

Figure 12: Examples of wheel-graphs.

In general the following holds.

Theorem 4.14. ([13]) The wheel-graph W_{2n+1} is non-word-representable for every $n \geq 2$.

We will show that the 5-wheel is non-word-representable in Theorem 4.40. In the same way one can prove that this theorem holds for all $n \geq 2$.

There are many more non-word-representable graphs. For instance, from [12] we know that there are 25 non-isomorphic non-word-representable graphs on 7 nodes, of which 15 contain the 5-wheel as an induced subgraph. There is, at this moment, no definitive answer to the question of whether there is a finite set of forbidden *induced* subgraphs[1]. Note that this would not result in an equivalent of Theorem 2.29, as this theorem states such a result for ordinary subgraphs.

Recently, there have been some results regarding representability of graphs via pattern-avoiding words, see [9, 15, 5], but these will not be discussed in this thesis.

4.3 Generic constructions

In Section 2.3 we talked about two kinds of orientations of graphs. In this section we will use these orientations to obtain representations for the graphs that admit them.

4.3.1 Transitive orientations

In this section we will use transitive orientations to obtain representations for certain graphs. This requires some additional terminology about orders and posets, which will be given here.

We start by repeating the definition of a transitive orientation.

Definition. ([7]) A directed graph $D = (W, A)$ is called *transitive* if the following property holds: If $(x, y) \in A$ and $(y, z) \in A$, then $(x, z) \in A$. An orientation $\mathcal{O}$ of a graph G is *transitive* if $G_{\mathcal{O}}$ is transitive.

Graphs that admit a transitive orientation, like bipartite graphs, are called *comparability graphs*. This family of graphs is important in the field of word-representable graphs, as it is equal to the family of permutationally representable graphs, see Definition 3.13. Before we prove this, we need some background in order theory.

Most of the following definitions originate from [16].

Remark 4.15. Given a set S and a binary relation $\mathcal{B} \subseteq S \times S$ we will write $x\mathcal{B}y$ instead of $(x, y) \in \mathcal{B}$.

Definition 4.16. A *strict partially ordered set*, or *strict poset*, is a pair $(S, <)$, where S is a set and $< \subseteq S \times S$ is a binary relation (the *partial order*) over the set S that satisfies the following:

- For all $x \in S, x \not< x$ (*irreflexivity*);
- If $x < y$ and $y < z$ then $x < z$ (*transitivity*);

[1] This is a set of graphs that may not be contained as induced subgraph.

- If $x < y$ then $y \not< x$ (*anti-symmetric*).

Definition 4.17. A *strict totally* or *linearly ordered set* is a pair $(S, <)$, where S is a set and $< \subseteq S \times S$ is a binary relation (the *total* or *linear order*), over the set S that satisfies the following:

- For all $x \in S, x \not< x$;
- If $x < y$ and $y < z$ then $x < z$;
- If $x < y$ then $x \not< y$;
- For all $x, y \in S$ with $x \neq y$ either $x < y$ or $y < x$.

In other words, a total order is a partial order in which every two elements are comparable. A finite linearly ordered set can be denoted by $(x_1, x_2, \dots, x_n)$ where $x_i < x_{i+1}$ for all $1 \leq i < n$.

Definition 4.18. A *linear extension* of a partial order $(S, <)$ is a total order $(S, <')$ such that $< \subseteq <'$ and $(S, <')$ is a total order.

Remark 4.19. Every finite poset has a linear extension, see [16].

Example 4.20. Let $S = \{1, 2, 3\}$ and $= \{(1, 2), (1, 3)\}$. We see that $(S, <)$ is a poset, but it is not a totally ordered set, as 2 and 3 are not comparable. If we add $(2, 3)$ to $<$, we get a totally ordered set $(S, < \cup \{(2, 3)\})$, which is a linear extension of $(S, <)$.

Definition 4.21. The *dimension* of a poset $(S, <)$ is the smallest number of linear orders $<_1, <_2, \dots, <_n$ of S such that $<= \cap_{i=1}^{n} <_i$. The set $\mathcal{R} = \{<_1, <_2, \dots, <_n\}$ is called a *realizer* of the poset.

Example 4.22. The poset in Example 4.20 has dimension 2, as it is not a totally ordered set, and the two linear orders $< \cup \{(2, 3)\}$ and $< \cup \{(3, 2)\}$ have $<$ as intersection.

Now we make a connection between comparability graphs and posets.

Lemma 4.23. Let G be a comparability graph and $\mathcal{O}$ a transitive orientation admitted by G. The the pair $G_{\mathcal{O}} = (W, A)$ is a strict poset.

Proof. It is trivial that $A \subseteq W \times W$.

Now let $x, y, x' \in W$. We know that $(x, x) \notin A$, as there are no arrows from a node to itself. If $(x, y) \in A$ and $(y, x') \in A$ we know that, because the orientation is transitive, $(x, x') \in A$. Lastly if $(x, y) \in A$ we know $(y, x) \notin A$, as two nodes can only have one arrow between them. This proves the properties stated in Definition 4.16, so (W, A) is a strict poset. □

We are now going to prove the following theorem.

Theorem 4.24. ([14]) A graph is permutationally representable if and only if it is a comparability graph.

Proof. For the forward direction, assume a graph G is permutationally represented by the t-uniform word w. For all the permutations $p_i(w)$ we can make a linear order $<_i$ by assuming that if x occurs before y in $p_i(w)$, then $x <_i y$.

It is easy to see that $x <_i y$ for all i or $y <_i x$ for all i if and only if x and y alternate in w. Also if $x <_i y$ and $y <_i x'$ for all i, then $x <_i x'$ for all i. If we now define an orientation $\mathcal{O}$ of G as $x \to y$ if and only if $x <_i y$ for all i, we obtain a transitive orientation that is admitted by G, thus G is a comparability graph.

For the converse, assume we have a comparability graph G and a transitive orientation $\mathcal{O}$ such that $G_{\mathcal{O}} = (W, A)$. Lemma 4.23 says that (W, A) is a strict poset. Assume $\mathcal{R} = \{<_1, <_2, \ldots, <_t\}$ is a realizer of the poset (W, A), and for every $1 \leq i \leq t$ define the word $w_i = x_{i,1}x_{i,2}\ldots x_{i,n}$ where $\#W = n$, and $x_{i,j} <_i x_{i,j+1}$ for all $1 \leq j < n$. Note that these words w_i are unique, as each $<_i$ is linear, which makes every two elements comparable. By the definition of a realizer, we know that $x < y$ if and only if $x <_i y$ for all i. This means that x and y alternate in $w = w_1 \ldots w_t$ if and only if $x < y$, and this holds if and only if $(x, y) \in A$. So the word w represents G and, as every w_i is a permutation, G is permutationally representable. □

We see that if we find the smallest permutation-representation of a graph, we have found a realizer of the poset associated with it. So if follows that the permutation-representation number of a comparability graph is equal to the dimension of the poset associated with it.

Remark 4.25. The permutation-representation number of a graph is not necessarily the same as the ordinary representation number. It is a easy to see that the permutation-representation number is at least as large as the representation number, but it can be larger, as we will show in the next example.

Example 4.26. We will take a look at the 3-cube, Q_3, see Figure 13.

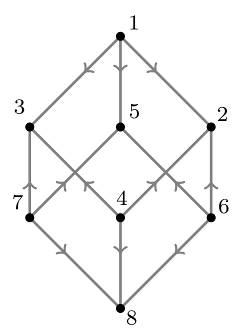


Figure 13: The 3-cube, Q_3, 3-represented by
$w = 567158372648123514736284$, with a transitive orientation.

In Theorem 5.8 we will show that Q_3 can be 3-represented by the word noted in Figure 13.

As Q_3 admits a transitive orientation, see Figure 13, we can use Theorem 4.24 to construct a permutation-representation for Q_3. The problem we would now face is finding a minimal realizer for the poset, as we want our word to be as small as possible. As this is a complex problem, see Remark 4.27, we assume we already have a realizer of this poset: $\mathcal{R} = \{(4, 6, 7, 8, 1, 2, 3, 5), (1, 4, 6, 2, 7, 3, 5, 8), (1, 4, 7, 3, 6, 2, 5, 8), (1, 6, 7, 5, 4, 2, 3, 8)\}$.

Using Theorem 4.24, the following word represents the graph $w =$ **46781235**14627358**14736258**16754238. For this graph it is provable by, for example, an exhaustive search that the permutation-representation number is not smaller than 4. This makes the permutation-representation number of this graph 4.

Remark 4.27. Theorem 4.24 reduces the problem of finding a permutation-representation of a graph to finding the dimension of the poset. Determining whether or not for a given $k \geq 3$ a poset has dimension at most k is an NP-complete problem, see [18]. In Section 4.4 we will talk about a general construction for permutation-representations of bipartite graphs, which allows us to find the mentioned 4-permutation-representation of Q_3.

4.3.2 Semi-transitive orientations

In this section we will focus on graphs that admit a semi-transitive orientation. Most of the theorems, lemmas and definitions in this section originate from [7], but with a different notation for orientations.

Semi-transitive orientations are at the core of word-representable graphs. We start again by repeating the definition.

Definition. A directed graph $D = (W, A)$ is called *semi-transitive* if the following properties hold:

- D acyclic;
- For every directed path $x_1, x_2, \ldots x_n$ in D one of the following two possibilities holds:
  - $(x_1, x_n) \notin A$;
  - $(x_1, x_n) \in A$ and $(x_i, x_j) \in A$ for all $1 \leq i < j \leq n$.

An orientation $\mathcal{O}$ of a graph G is *semi-transitive* if $G_{\mathcal{O}}$ is semi-transitive.

To make the upcoming lemmas easier to prove, we will first rephrase the notion a of semi-transitive orientation. For this we will use the following definitions and lemma.

Definition 4.28. A *semi-cycle* is an acyclic directed graph $D = (W, A)$ with $\#W \geq 2$ with $W = \{x_1, x_2, \ldots, x_n\}$, and $A = \{(x_1, x_2), (x_2, x_3), \ldots, (x_{n-1}, x_n), (x_1, x_n)\}$.

We observe that a semi-cycle can be obtained by reversing the orientation of one edge in a cyclically oriented cycle-graph.

Definition 4.29. A directed graph $D = (W, A)$ is a *shortcut* if the following properties hold:

- $\#W \geq 4$;
- D is acyclic;
- There are nodes $x, y \in W$ such that there is no arrow between x and y;
- There is a subgraph $H = (W_H, A_H)$ of D such that H is a semi-cycle and $W_H = W$.

A directed graph D *contains a shortcut* if there is a subgraph of D that is a shortcut.

Lemma 4.30. An orientation $\mathcal{O}$ of an undirected graph G is semi-transitive if and only if $G_{\mathcal{O}}$ is acyclic and does not contain any shortcuts.

Proof. Assume we have a semi-transitive orientation $\mathcal{O}$ of a graph G. By definition, $G_{\mathcal{O}} = (W, A)$ is acyclic.

Now assume $G_{\mathcal{O}}$ contains a shortcut with nodes $x_1, x_2, \ldots, x_n$ where $n \geq 4$. So without loss of generality we may assume there is a directed path $x_1, x_2, \ldots, x_n$ and there is an arrow from $(x_1, x_n) \in A$. As $\mathcal{O}$ is semi-transitive and $(x_1, x_n) \in A$, we know that for all $1 \leq i < j \leq n$ we have $(x_i, x_j) \in A$, which cannot be the case when this is a shortcut. So $G_{\mathcal{O}}$ is acyclic and does not contain any shortcuts.

Assume $G_{\mathcal{O}}$ is acyclic and does not contain any shortcuts. Also assume there is a directed path $x_1, x_2, \ldots, x_n$ in $G_{\mathcal{O}}$. If $(x_1, x_n) \notin A$, we are done, so assume $(x_1, x_n) \in A$. In this case the nodes $x_1, \ldots, x_n$ form a semi-cycle. As $G_{\mathcal{O}}$ does not contain any shortcuts, we know that for every $1 \leq i < j \leq n$ it holds that $(x_i, x_j) \in A$ or $(x_j, x_i) \in A$. As $G_{\mathcal{O}}$ is acyclic, we know that $(x_i, x_j) \in A$, proving that $\mathcal{O}$ is semi-transitive. $\square$

We will now formulate one of the most important results regarding word-representable graphs. As it turns out, the family of word-representable graphs is equal to the family of graphs that admit a semi-transitive orientation. We will not prove this, but we will use the constructions in the proof to compare later results to. First we need some more terminology.

Definition 4.31. For a directed graph $D = (W, A)$, a permutation π of W is called a *topsort* if for all $x, y \in W$, when there is a directed path from x to y, then x is to the left of y in π.

Definition 4.32. For a directed graph $D = (W, A)$, a word w *covers* a set $N \subseteq A^C$ of non-edges of D if:

- w is k-uniform for some k;
- $p(w)$ is a topsort of D;
- D is a subgraph of $D(w)$;
- $N \subseteq D(w)^C$.

Definition 4.33. For a directed graph $D = (W, A)$ and $x \in W$ we distinguish five sets of nodes:

- $I(x) = \{y \in W \mid (y, x) \in A\}$;
- $A(x) = \{y \in W \mid y \rightsquigarrow x\} \setminus I(x)$;
- $O(x) = \{y \in W \mid (x, y) \in A\}$;
- $B(x) = \{y \in W \mid x \rightsquigarrow y\} \setminus O(x)$;
- $R(x) = W \setminus (\{x\} \cup I(x) \cup O(x) \cup A(x) \cup B(x))$.

With these definitions we can state one of the main results of [7].

Lemma 4.34. Let $G = (V, E)$ be a graph that admits a semi-transitive orientation $\mathcal{O}$ and let $x \in V$. Then the non-edges incident to x can be covered by a 2-uniform word.

Remark 4.35. This lemma does not give us a construction immediately, but the proof given in [7] does. We can state this as follows: The non-edges incident to x can be covered by the 2-uniform word $w_x = A\ I\ R\ A\ x\ O\ I\ x\ B\ R\ O\ B$, where A, I, B, O and R are topsorts of the sets $A(x), I(x), B(x), O(x)$ and $R(x)$ of $G_{\mathcal{O}}$ respectively.

The proof of the following theorem relies extensively on Lemma 4.34. Note that a *clique* of a graph G is a subgraph of G that is complete.

Theorem 4.36. A graph G is word-representable if and only if it admits a semi-transitive orientation. Moreover, each non-complete word-representable graph is $2(n-\kappa)$-word-representable where κ is the size of the maximum clique in G.

Remark 4.37. For the construction of a word-representant for such a graph, we need to consult the proof of this theorem. The construction can be stated as follows: For every node x that is not part of the maximum clique of G, determine w_x using Lemma 4.34. Concatenating these w_x for all such x, we obtain a word that $2(n-\kappa)$-represents G.

Remark 4.38. The use of the maximum clique in Theorem 4.36 is not needed but makes the word we find shorter. This works because no node in the maximum clique has any non-edges that are not already covered by a node outside of the maximum clique. Stated differently, adding w_x for some x in the maximum clique does not cover any non-edges that would not be covered otherwise.

Also, this construction gives us a uniform word representing the graph, but not necessarily the shortest one, which we will show in the next example. In general this construction will give a k-uniform word where k is much larger than the representation number.

Example 4.39. We will use the 3-cube and its semi-transitive orientation from Figure 13. We start by identifying the maximum clique in Q_3. This can be any two connected nodes, as the maximum clique has size two. We will use $\{7, 8\}$ as the nodes that form the maximum clique.

For every node in $W \setminus \{7, 8\}$ we will construct a word as in Remark 4.35, see Table 1.

x	I	A	O	B	T	w_x
1	ϵ	ϵ	235	ϵ	467**8**	4678123514678235
2	146	ϵ	ϵ	ϵ	**7**358	1467358214627358
3	147	ϵ	ϵ	ϵ	**6**258	1476258314736258
4	ϵ	ϵ	238	ϵ	167**5**	1675423841675238
5	167	ϵ	ϵ	ϵ	**4**238	1674238516754238
6	ϵ	ϵ	258	ϵ	147**3**	1473625861473258

Table 1: Topsorts of the different neighbour sets for all nodes not in the maximum clique of Q_3.

Concatenating all these words gives the following 12-uniform word:
$w =$ 4678123514678235**1467358214627358**1476258314736258**1675423841675238**1674238516754238**1473625861473258**. We already saw in Example 4.26 that this graph is 4-permutation-representable, thus this construction does not give the shortest word.

Now we know the main criterion for a graph to be word-representable, we can prove that the 5-wheel from Example 4.13 is not word-representable by showing it does not admit a semi-transitive orientation.

We will do this by looking at all possible ways of orienting edges in the graph to obtain a semi-transitive orientation. For example, Figure 14 shows the only ways to complete a partially orientated graph without creating a shortcut or cycle, which are very important in the following proof.

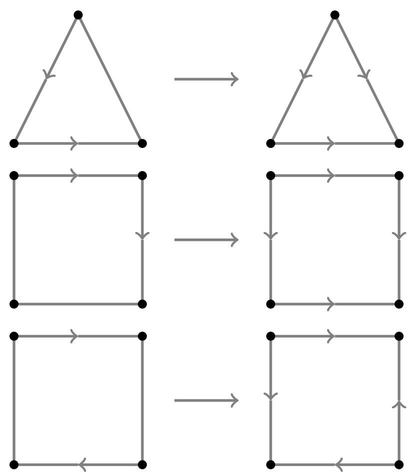


Figure 14: The only possibilities in completing the orientation of a 3- and 4-cycle.

Theorem 4.40. The 5-wheel does not admit a semi-transitive orientation.

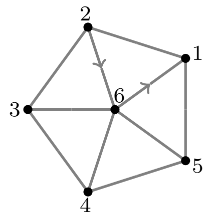
Proof. We start by noting that we cannot orient all the edges incident to node 6 outwards or inwards at the same time without necessarily creating a shortcut when orienting the remainder of the edges. This implies that there is at least one outgoing arrow and one incoming arrow which, without loss of generality, gives us graph A of Figure 16.

From Figure 14 we know that the only way to semi-transitively complete the triangle 1,2,6 is shown in graph B of Figure 14.

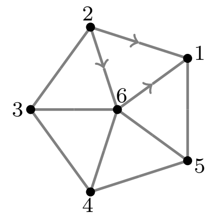
At this point we have to choose an edge we will orient. As there are no restrictions on which edge to choose, we choose the edge $\{2, 3\}$, giving us two possibilities, see graphs $C1$ and $C2$. Graph $C1$ can only be completed as graph $D1$, as not to have a shortcut on nodes 1, 2, 3 and 6. Graph $C2$ can only be completed as graph $D2$, as not to have a cycle on nodes 2, 3 and 6.

The remainder of the steps are shown in Figure 16. Graphs that have blue nodes indicate a shortcut and thus the end of that search-branch. Every change in letter indicates an extra oriented edge and an extra number indicates a choice for an edge to orient.

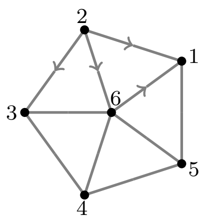
This shows that the 5-wheel does not admit a semi-transitive orientation. □



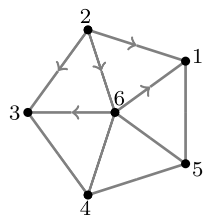
A



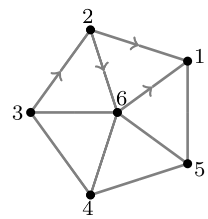
B



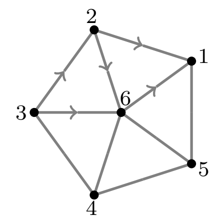
*C*1



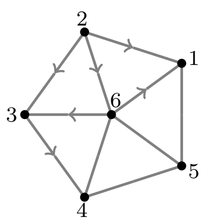
*D*1



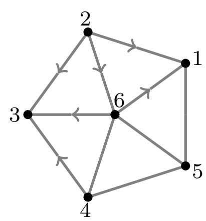
*C*2



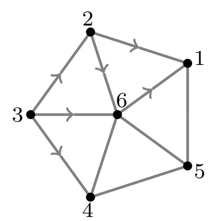
*D*2



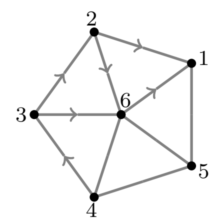
*E*1.1



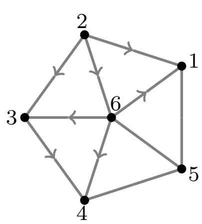
*E*1.2



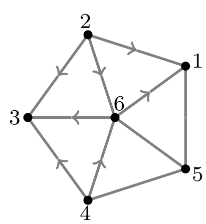
*E*2.1



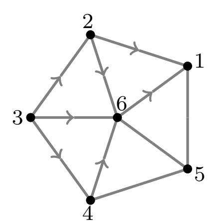
*E*2.2



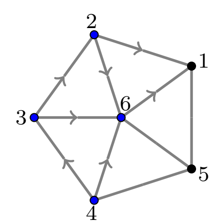
*F*1.1



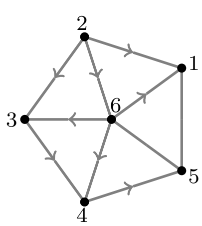
*F*1.2



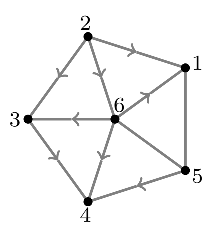
*F*2.1



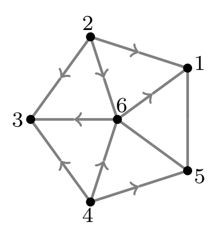
*F*2.2



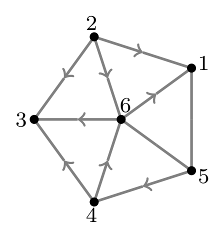
*G*1.1.1



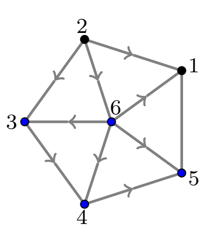
*G*1.1.2



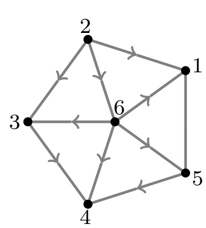
*G*1.2.1



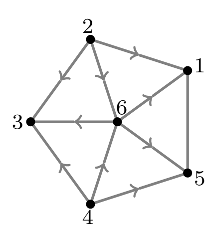
*G*1.2.2



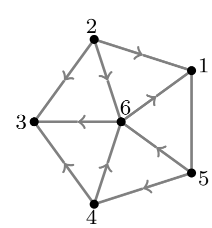
*H*1.1.1



*H*1.1.2



*H*1.2.1



*H*1.2.2

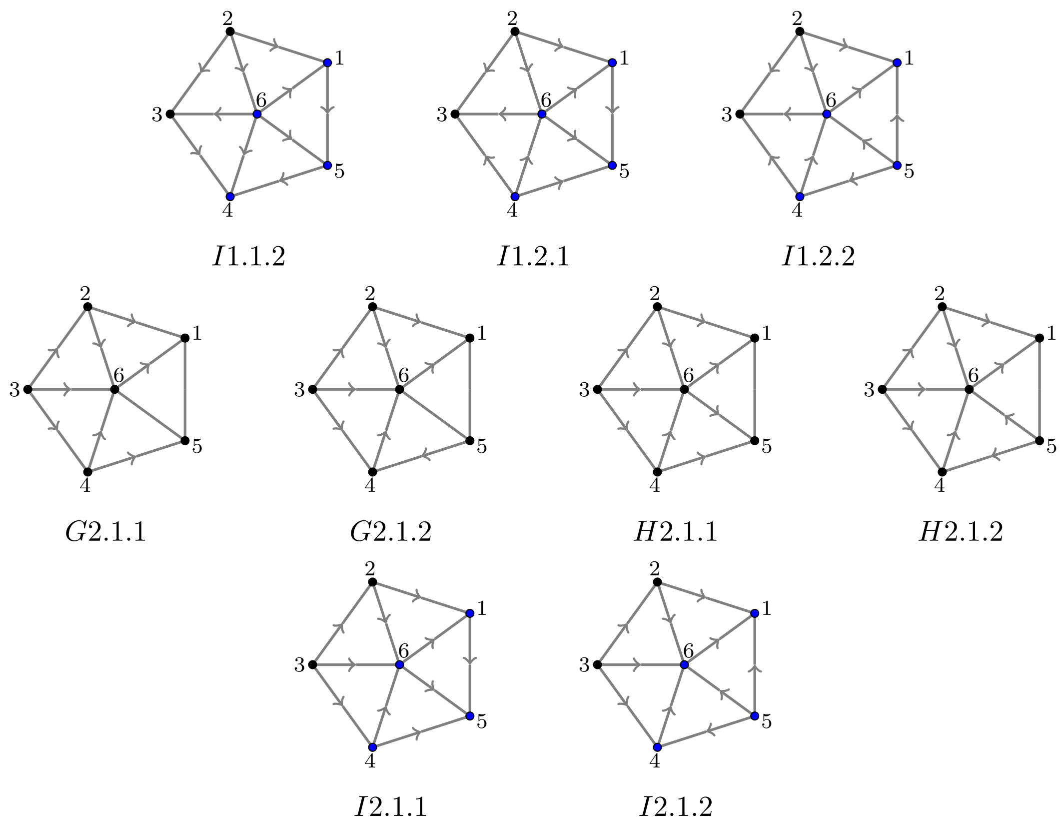


Figure 16: Figures illustrating the proof of Theorem 4.40.

4.4 Bipartite graphs

In this section we will look at the construction of a permutation-representation for bipartite graphs. This appears to be a new result that gives a permutation-representation and an upper bound on the permutation-representation number of a bipartite graph. To be able to do this for every bipartite graph we use the following lemmas.

Lemma 4.41. Let $G = (V, E)$ and $H = (V', E')$ be graphs with $V \cap V' = \emptyset$ that can be permutationally represented by $w = p_1 p_2 \dots p_k$ and $w' = q_1 q_2 \dots q_l$ respectively, where $2 \leq k \leq l$. Then the union of G and H can be permutationally represented by
$v = p_1 q_1 p_2 q_2 \dots p_{k-1} q_{k-1} q_k p_k q_{k+1} p_k \dots q_l p_k$.

Proof. It suffices to prove the following:

- v is a concatenation of permutations of $V \cup V'$;
- v_V represents G;
- $v_{V'}$ represents H;
- For all $x \in V$ and $y \in V'$ it holds that they do not alternate in v.

As for every i the words p_i and q_i are permutations of V and V' respectively, for each i and j the words $p_i q_j$ and $q_j p_i$ are permutations of $V \cup V'$, which makes v a concatenation of permutations of $V \cup V'$.

We see that $v_V = p_1 p_2 \dots p_k \dots p_k = w p_k \dots p_k$. By Lemma 3.23 with $A_1 = N_k$ and $A_2 = \dots = A_{l-k} = \{k\}$ it follows that v_V represents G.

Also we see that $v_{V'} = q_1 q_2 \dots q_l = w'$, which represents H.

Let $x \in V$ and $y \in V'$. We see that $v_{\{x,y\}} = (xy)^{k-1}(yx)^{l-k+1}$. As $k \leq l$ and $k \geq 2$ it follows that $l - k + 1 > 0$ and thus x and y do not alternate in v. □

Lemma 4.42. Let $G = (V, E)$ be a non-directed graph that can be k-represented by w, with $2 \leq k$, and let $y \in V$. Now let $G' = (V', E')$ be the graph with $V' = V \cup \{y'\}$ and $E' = E \cup \{\{y', x\} \mid x \in N(y)\}$. Then G' can be k-represented by $f(w)$, where

$$f(x, i) = \begin{cases} x & \text{if } x \neq y \\ y'y & \text{if } i = 1 \text{ and } x = y \\ yy' & \text{otherwise.} \end{cases}$$

The proof of this lemma omitted here, as it should be straightforward.

Definition 4.43. Given a graph $G = (V, E)$ and a node $x \in V$ we define the word w_x as the word obtained by concatenating all letters in $N(x)$ such that y appears left of y' if $\#N(y) < \#N(y')$ and ties may be broken in any way.

Proposition 4.44. Let $G = (V, E)$ be a connected bipartite graph with parts $V = A \cup B$ where no two nodes have the same neighbourhood and let $A = \{a_1, a_2, \dots, a_n\}$. Then G has an n-permutation-representant given by
$w = f(g(a_1 a_2 \dots a_n)) f(g(a_2 a_3 \dots a_n a_1)) \dots f(g(a_n a_1 \dots a_{n-1}))$, where

$$f(x, i) = \begin{cases} x & \text{if } i = 1 \\ \epsilon & \text{otherwise} \end{cases} \qquad g(x, i) = \; w_x x.$$

Proof. It suffices to prove the following:

- w is a concatenation of n permutations of $A \cup B$;
- $x, y \in A$ do not alternate in w;
- $x, y \in B$ do not alternate in w;
- $x \in A$ and $y \in B$ alternate in w if and only if $\{x, y\} \in E$.

As G is connected, every $b \in B$ is in $N(a)$ for some $a \in A$. This implies that $g(a_i \dots a_{i-1})$ contains every $x \in A \cup B$ for all i and thus, by definition of f, $f(g(a_i \dots a_{i-1}))$ is a permutation of $A \cup B$ for every i. This shows that w is a concatenation of n permutations.

Let $a_i, a_j \in A$ and assume without loss of generality that $i < j$. We see that in $f(g(a_i a_{i+1} \dots a_{i-1}))$, a_i appears left of a_j, while in $f(g(a_{i+1} a_{i+2} \dots a_i))$ it appears right of a_j. This implies that a_i and a_j do not alternate in w.

Let $x, y \in B$ with $N(x) \neq (N(x) \cap N(y)) \neq N(y)$. This means there is an $a_i \in N(x) \setminus N(y)$ and an $a_j \in N(y) \setminus N(x)$. We see that in $f(g(a_i \dots a_{i-1}))$, x appears left of y, while in $f(g(a_j \dots a_{j-1}))$ it appears right of y. This implies that x and y do not alternate in w.

Let $x, y \in B$ with $N(y) \subset N(x)$. As $N(y) \neq N(x)$, there is an $a_i \in N(x) \setminus N(y)$. We see that in $f(g(a_i \dots a_{i-1}))$, x appears left of y. Also we know that $N(y) \neq \emptyset$, and for every $a_j \in N(y)$ we see that y appears left of x in $f(g(a_j \dots a_{j-1}))$ as $\#N(y) < \#N(x)$. This implies that x and y do not alternate in w.

Let $a \in A$, $b \in B$ and $\{a, b\} \in E$, i.e. $b \in N(a)$. We see that, by construction, b will appear at least one time to the left of a in $g(a_i \dots a_{i-1})$ for all i. As a occurs exactly once in this word, by construction we have for all i that b appears left of a in $f(g(a_i \dots a_{i-1}))$. This implies that a and b alternate in w.

Let $a_i \in A$, $b \in B$ and $\{a_i, b\} \notin E$, i.e. $b \notin N(a_i)$. We see that a is the last letter of the word $f(g(a_{i+1} \dots a_i))$ and thus b appears left of a. As $b \notin N(a_i)$, we see that b does not appear left of a_i in $g(a_i \dots a_{i-1})$ and thus appears right of a_i in $f(g(a_i \dots a_{i-1}))$. This implies that a_i and b do not alternate in w. □

Theorem 4.45. The permutation-representation number of any bipartite graph $G = (V, E)$ with parts $V = A \cup B$ is at most $\min\{\#\{N(a) \mid a \in A\}, \#\{N(b) \mid b \in B\}\}$.

Proof. By Lemma 4.41 we may assume G to be connected, as the permutation-representation number of the union of two bipartite graphs is at most the maximum of the two individual parts.

From Lemma 4.42 it follows that in order to permutationally represent every connected bipartite graph, it suffices to be able to construct a permutation-representation for bipartite graphs where no two nodes have the same neighbourhood. Let $G' = (V', E')$ be an induced subgraph of G such that for all $x, y \in V'$ we have $N(x) \neq N(y)$ and $\{N(x) \mid x \in V\} = \{N(x) \mid x \in V'\}$. We can now apply Proposition 4.44 to find a $\min\{\#\{N(a) \mid a \in A\}, \#\{N(b) \mid b \in B\}\}$-permutation-representation for G'. With Lemma 4.42 we can now add the nodes in $V \setminus V'$ to G' to obtain a $\min\{\#\{N(a) \mid a \in A\}, \#\{N(b) \mid b \in B\}\}$-permutation-representation for G. □

Remark 4.46. At this point in time it is not clear whether or not this is a completely new result, as it could be equivalent to known results in the field of order theory. We will discuss some open questions that arise from this in Chapter 6.

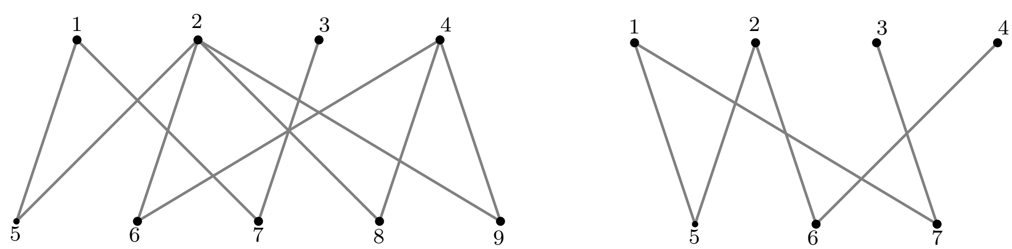
Example 4.47. We will take a look at the bipartite graph G in Figure 17a. We see that nodes 6, 8 and 9 have the same neighbourhoods, so for now we will ignore nodes 8 and 9 and focus on the *reduced* version of G in Figure 17b. To make the shortest representation we will choose $A = \{5, 6, 7\}$ and $B = \{1, 2, 3, 4\}$.

Following Proposition 4.44 we find the following:

- $n = 3$ and $a_1a_2a_3 = 567$;
- $w_5 = 12$, $w_6 = 42$, $w_7 = 31$;
- $g(a_1a_2a_3) = 125426317$, $g(a_2a_3a_1) = 426317125$ and $g(a_3a_1a_2) = 317125426$;
- $f(g(a_1a_2a_3)) = 1254637$, $f(g(a_2a_3a_1)) = 4263175$ and $f(g(a_3a_1a_2)) = 3172546$,

were we have broken ties by ordering by increasing numeric value.

So the word $w = 1254637\mathbf{4263175}3172546$ represents the graph in Figure 17b. Now, using Lemma 4.42, we can add nodes 8 and 9 to obtain a representation for the graph in Figure 17a, which results in $v = 125489637\mathbf{426983175}317254698$ as 3-permutation-representation.



(a) A bipartite graph G. (b) Reduced version of G.

Figure 17: A bipartite graph and its reduced version.

Remark 4.48. In Example 4.47 we chose A and B such that we would obtain a smaller word. We could have chosen A and B the other way around, but this would have given us the 4-permutation-representation $v = 5716234\mathbf{5627341}7364512\mathbf{6457123}$, instead of a 3-permutation-representation. As Proposition 4.44 results in a $\#\{N(a) \mid a \in A\}$-permutation representation, choosing A different can change the length of the found representation.

4.5 Existing constructions

In this section we will look at some existing constructions of words for graphs. Most of the constructions used in this section originate from [12], but have been rephrased using occurrence-based functions where possible. For each of these constructions we will give the construction and an example, but not a proof.

Note that there are many more constructions known, for example see [6], but not all can be incorporated here.

4.5.1 Trees

The trees are a basic family of graphs. With the following construction it is possible to represent trees, and with that forests (set of independent trees), but we will focus on trees.

Definition 4.49. A graph $G = (V, E)$ is a *tree* if and only if for every two nodes x and y there is exactly one path from x to y.

Construction 4.50. The following algorithm gives a 2-representation w for a tree $G = (V, E)$:

1. $w = 11$, $C = \emptyset$;
2. While $C \neq V$ do the following:

   2.1 $A = \{x \in V \setminus C \mid w \text{ contains } x\}$;

   2.2 $B = \{\{a, y\} \in E \mid a \in A, y \in V \setminus (A \cup C)\}$;

   2.3 For all $\{a, y\} \in B$ replace w by $w = h_{a,y}(w)$, where

$$h_{a,y}(x, i) = \begin{cases} yxy & \text{if } a = x \text{ and } i = 2 \\ x & \text{otherwise;} \end{cases}$$

   2.4 Replace $C = C \cup A$.

Example 4.51. We will represent the tree in Figure 18.

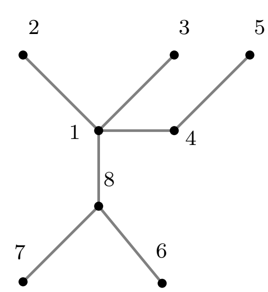


Figure 18: A tree.

We can do steps 1, 2.1 and 2.2 all at once. This gives $w = 11$, $A = \{1\}$, $B = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 8\}\}$ and $C = \emptyset$. So in 2.3 and 2.4 we get:

- $h_{1,2}(w) = 1212$;
- $h_{1,3}(1212) = 123132$;
- $h_{1,4}(123132) = 12341432$;
- $h_{1,8}(12341432) = 1234818432$;
- $C = \{1\}$.

Since $C \neq V$ we have to go on and get $A = \{2, 3, 4, 8\}$, $B = \{\{4, 5\}, \{8, 6\}, \{8, 7\}\}$. After 2.4 this gives $w = h_{8,7}(h_{8,6}(h_{4,5}(w))) = 1234816787654532$ and $C = \{1, 2, 3, 4, 8\}$.

After this we still need to go on, but we have already processed all edges so we will see that $B = \emptyset$ in this step and it results in $C = V$. The word $w = 1234816787654532$ represents the graph.

Remark 4.52. As a tree does not contain any cycles, the maximum clique we can find in a tree has size two. This means that Theorem 4.36 gives a $2(n-2)$-representation for a tree with n nodes, while we find a 2-representation for every tree with Construction 4.50.

4.5.2 Cycles

Recall the definition of a cycle graph.

Definition. A graph $G = (V, E)$ is a *cycle graph* if $V = \{x_1, x_2, \ldots, x_n\}$ for some n, and $E = \{\{x_1, x_2\}, \{x_2, x_3\}, \ldots, \{x_{n-1}, x_n\}, \{x_1, x_n\}\}$.

In the construction of a word for cycles we will use Construction 4.50.

Remark 4.53. The graphs C_1 and C_2 are the complete graphs on 1 and 2 nodes respectively, and thus can be easily 1-represented. In the upcoming construction we will only look at the more interesting cases of C_n with $n \geq 3$.

Construction 4.54. The following construction gives a 2-representation w for the cycle graphs $C_n = (V, E)$ with $n > 2$:

1. Apply Construction 4.50 to the tree obtained by removing the edge $\{1, n\}$ from the graph, and obtain the word w';

2. Now the word $w = h(w')$ represents the graph C_n, where

$$h(x, i) = \begin{cases} xn & \text{if } x = 1, i = 1 \\ \epsilon & \text{if } x = n, i = 2 \\ x & \text{otherwise.} \end{cases}$$

Example 4.55. We will construct a representant for the 6-cycle, shown in Figure 19. Step 1 gives us the word $w' = 121324354656$ with Construction 4.50. The word $w = h(w') = 162132435465$ now is a 2-representant for C_6.

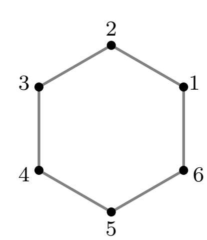


Figure 19: The 6-cycle C_6.

Remark 4.56. For the graph C_3, which is equivalent to the complete graph with 3 nodes, we find a 2-representation while this graph can be easily 1-represented. As C_n is not equivalent to a complete graph for $n > 3$, is it not 1-representable. We have found a 2-representation, so this must be optimal.

Theorem 4.36 gives a $2(n-2)$-representation for cycle graphs because, as was the case with trees, the largest clique in a cycle has size 2.

4.5.3 Ladders

The construction of words for the *ladder graphs* will be used to create words for prisms later on.

Definition 4.57. ([13]) For $n \geq 2$ the ladder graph L_n is the graph that has $V = \{1, 2, \ldots, n, 1', 2', \ldots, n'\}$ and $E = \{\{i, i+1\} \mid 1 \leq i < n\} \cup \{\{i', (i+1)'\} \mid 1 \leq i < n\} \cup \{\{i, i'\} \mid 1 \leq i < n\}$.

Construction 4.58. ([13]) The following construction gives a 3-representation w for the ladder graphs $L_n = (V, E)$ for $n \geq 2$:

1. Start with $w = 121'12'21'2'11'22'$, 3-representing L_2;
2. Now for all $3 \leq j \leq n$ replace $w = h_j(w)$, where

$$h_j(x, i) = \begin{cases} jj'xj & \text{if } x = j-1 \text{ and } i = 1 \\ j'xjj' & \text{if } x = (j-1)' \text{ and } i = 2 \\ x & \text{else} \end{cases}$$

   which results in a 3-representation for L_n.

Example 4.59. We will now construct a representant for the 4-ladder shown in Figure 20.

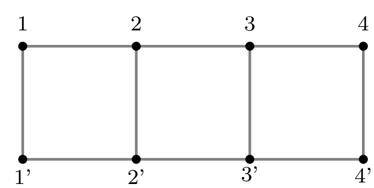


Figure 20: The 4-ladder L_4.

We start with $w = 121'12'21'2'11'22'$. In step 2 we get

- $w = h_3(w) = 1\mathbf{33'23}1'12'21'\mathbf{3'2'33'}11'22'$;
- $w = h_4(w) = 1\mathbf{44'34}3'231'12'21'\mathbf{4'3'44'}2'33'11'22'$,

where the **bold** subwords are the substituted parts.
The word $w = 144'343'231'12'21'4'3'44'2'33'11'22'$ is a 3-representant of L_4.

Remark 4.60. The ladder L_n has $2n$ nodes, so Theorem 4.36 gives a $4(n-1)$-representation for L_n, as the largest cycle has size 2.

4.5.4 Prisms

For now we will use the following construction to represent prisms. In Chapter 5 we will develop a new way of representing them.

Definition 4.61. ([2]) For $n \geq 3$ the n-prism, Pr_n, is the graph with $V = \{1, 2, \ldots, n, 1', \ldots, n'\}$ and $E = \{\{i, j\} \,|\, 1 \leq i < n, j = i + 1\} \cup \{\{i', j'\} \,|\, 1 \leq i < n, j = i + 1\} \cup \{\{i, i'\} \,|\, 1 \leq i \leq n\} \cup \{\{1, n\}, \{1', n'\}\}$.

This definition is the same as saying that the sets $\{1, 2, \ldots, n\}$ and $\{1', 2', \ldots, n'\}$ form cycles and there are the additional edges $\{i, i'\}$ for $1 \leq i \leq n$.

Construction 4.62. The following construction gives a 3-representation w for the prisms $Pr_n = (V, E)$:

1. Apply Construction 4.58 to the ladder obtained by removing the edges $\{1, n\}$ and $\{1', n'\}$ from the graph, and obtain the word w';
2. Now the word $w = h(w')$ represents the graph Pr_n, where

$$h(x, i) = \begin{cases} n & \text{if } x = 1, i = 1 \\ 1 & \text{if } x = n, i = 1 \\ n' & \text{if } x = 1, i = 2 \\ 1' & \text{if } x = n, i = 2 \\ x & \text{otherwise.} \end{cases}$$

Example 4.63. We will now make a representant for the 4-prism, shown in Figure 21.

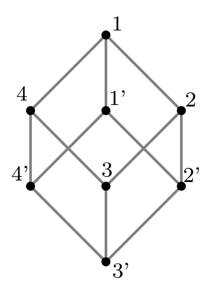


Figure 21: The 4-prism Pr_4.

Step 1 gives us the word $w' = 144'343'231'12'21'4'3'44'2'33'11'22'$, as shown in Example 4.59. The word $w = h(w') = w = \mathbf{41}4'343'231'12'2\mathbf{4'1'}3'44'2'33'11'22'$ now is a 3-representant for Pr_3, where the **bold** subwords are again the substituted parts.

Remark 4.64. The 4-prism is equivalent to the 3-cube, Q_3, of which we have already seen a different 3-representation.

Remark 4.65. The prism Pr_n contains $2n$ nodes and the largest clique has size 3 if $n = 3$, and size 2 in all other cases. This means that Theorem 4.36 gives a 6-representation for the 3-prism and a $4(n-1)$-representation for all other prisms.

5 Cartesian product

In Section 4.1 we talked about some operations on graphs and how these affect the representability of the graph. In this chapter we will focus on a more complex operation, the Cartesian product of two graphs, and prove some new results.

As part of this research, a small Java-tool was developed to perform the upcoming constructions easily and produce LaTeX-code for the graphical representations of the resulting graphs. This tool is available on request by sending an e-mail to the author.

As we have seen before, if a non-word-representable graph H is an induced subgraph of a graph G, then G is also non-word-representable. This implies that the Cartesian product of a non-word-representable graph with another graph will be non-word-representable. As it turns out, the following theorem holds.

Theorem 5.1. ([12]) The Cartesian product $G \,\square\, H$ is word-representable if and only if both G and H are word-representable.

Remark 5.2. Although this theorem shows us that $G \,\square\, H$ is word-representable, it does not give a construction of a word that represents this graph, as it is based on finding a semi-transitive orientation for the result.

To find a word-representation, one could use Theorem 4.36 to construct a word from this semi-transitive orientation, but as we mentioned before, this likely results in a non-optimal word.

The remainder of this chapter will be focused on finding more optimal words for representing $G \,\square\, H$ for special cases of H. It is structured in the same order as the results were developed, getting more general along the way.

5.1 Cartesian product with K_2

The first special case we will take a look at is the Cartesian product of a graph G with the complete graph on 2 nodes, K_2.

5.1.1 Representation

When looking at the Cartesian product of a graph G and K_n with $n > 1$, the resulting graph consists of n copies of G, in which moreover any two nodes corresponding to the same node in G are connected by an edge, as was illustrated in Figure 4. Also, remember that the complete graph on n nodes is represented by the 1-uniform word $w = 12\ldots n$.

The complete graph K_2 consists of two nodes that are connected by a single edge. The nodes of $G \,\square\, K_2$ where $G = (V, E)$ are denoted by x_1, x_2 for all $x \in V$. Two nodes x_i, y_j are connected by an edge in $G \,\square\, K_2$ if and only if

- $i = j$ and $(x, y) \in E$, or
- $i \neq j$ and $x = y$.

We write $V_1 = \{x_1 \,|\, x \in V\}$ and $V_2 = \{x_2 \,|\, x \in V\}$, so $V_1 \cup V_2$ is the set of nodes of $G \,\square\, K_2$.

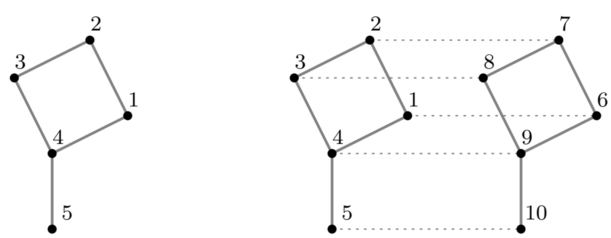
Now we will state the main result of this section. We will not prove this, because it is a special case of Theorem 5.14, which we will prove in the next section.

Theorem 5.3. Let G be a k-representable graph for $k > 1$ and let w be a k-representant of G. Then the graph $G \,\square\, K_2$ is $(k+1)$-representable with representant $w' = f(w)g(w)$ for the occurrence based functions f, g defined by

$$f(x,i) = \begin{cases} x_1 & \text{if } i = 1 \\ x_2x_1 & \text{if } 1 < i \leq k, \end{cases} \qquad g(x,i) = \begin{cases} x_2 & \text{if } i = 1 \\ x_1x_2 & \text{if } i = 2 \\ \epsilon & \text{if } 2 < i \leq k. \end{cases}$$

Remark 5.4. This theorem gives a word consisting of letters in $V_1 \cup V_2$. If necessary, one can rename the nodes of the resulting graph to make this construction repeatable.

Example 5.5. We will look at the Cartesian product of the graph G from Figure 22a, represented by the word $w = 3142132545$, and K_2.



(a) The graph G. (b) $G \,\square\, K_2$.

Figure 22: Example of Cartesian product with K_2.

When we follow the construction of Theorem 5.3 we find:

- $f(w) = 3_1 1_1 4_1 2_1 1_2 1_1 3_2 3_1 2_2 2_1 5_1 4_2 4_1 5_2 5_1$;
- $g(w) = 3_2 1_2 4_2 2_2 1_1 1_2 3_1 3_2 2_1 2_2 5_2 4_1 4_2 5_1 5_2$.

When we concatenate these words and rename every letter x_i to $x + (i-1) \cdot 5$, we get the word $w' = 3142618372594(10)5$**8697163827(10)495(10)**, which represents the Cartesian product displayed in Figure 22b.

Theorem 5.3 also implies the following.

Corollary 5.6. Let G be a graph with representation number $k \geq 2$, then the graph $G \,\square\, K_2$ has representation number k or $k+1$.

Remark 5.7. Theorem 5.3 and Corollary 5.6 give rise to a couple of questions, such as: When is this construction optimal? We formulated some of these questions in Chapter 6.

5.1.2 Cubes and prisms

Theorem 5.3 is a result of an attempt to prove that the k-cube, Q_k, has representation number k. To prove that a graph has representation number k, one needs to prove that there is a k-representation, and that there is no l-representation with $l < k$. The first part of proving this for the Q_k is a direct consequence of Theorem 5.3 while the other part remains as an open question.

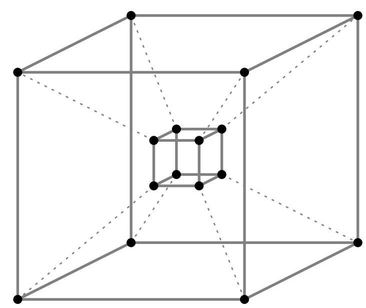


Figure 23: The graph Q_4 as Cartesian product of Q_3 and K_2.

The observation that led to this construction is that Q_k is the Cartesian product of Q_{k-1} and K_2, see Figure 23. Together this led to the following theorem.

Theorem 5.8. For every $k \geq 1$, Q_k is k-representable.

Proof. The proof is done by induction on k. For $k = 1$ we observe that Q_1 is the same graph as K_2, which can be 1-represented by the word $w = 12$. For $k = 2$ we observe that Q_2 is the same graph as a 4-cycle, which can be 2-represented by the word $w = 31421324$, see Construction 4.54.

For the induction step, we use that the Cartesian product $Q_{k-1} \Box K_2$ forms a k-cube and we assume to have a $(k-1)$-uniform representant of Q_{k-1}. Now using Theorem 5.3 we construct a k-uniform representant for the k-cube from the $(k-1)$-uniform representant of the $(k-1)$-cube. □

Remark 5.9. As the maximum clique in Q_k is of size 2 and Q_k has 2^k nodes, Theorem 4.36 results in a $2(2^k - 2) = (2^{k+1} - 4)$-uniform word for Q_k, while Theorem 5.3 results in a k-uniform word. For Q_3 this is already a factor 4 shorter.

Example 5.10. Table 2 gives an overview of uniform words for several cubes.

Theorem 5.3 also implies some results about prisms, as they are the Cartesian product of a cycle graph and K_2.

Corollary 5.11. All prisms are have representation number 3.

From Construction 4.54 we know that all cycle-graphs are 2-representable. As a prism is the Cartesian product of a cycle-graph and K_2, Theorem 5.3 implies that every prism is 3-representable. It was first proven in [11] that prisms are not 2-representable which, together with this result, implies that the representation number of prisms must be 3.

k	k-uniform word
1	12
2	31421324
3	314251736284758615372648
4	31425917(11)36(10)28(12)4(15)7(13)5(16)8(14)691(13)5(11)3 (15)7(10)2(14)6(12)4(16)8(11)9(12)(10)(13)19(15)3(11)(14) 2(10)(16)4(12)7(15)5(13)8(16)6(14)
5	314259(17)17(11)(19)36(10)(18)28(12)(20)4(15)(23)7(13)(21) 5(16)(24)8(14)(22)6(25)9(17)1(29)(13)(21)5(27)(11)(19)3(31) (15)(23)7(26)(10)(18)2(30)(14)(22)6(28)(12)(20)4(32)(16)(24) 8(27)(11)(25)9(28)(12)(26)(10)(29)(13)(17)1(25)9(31)(15)(19) 3(27)(11)(30)(14)(18)2(26)(10)(32)(16)(20)4(28)(12)(23)7(31) (15)(21)5(29)(13)(24)8(32)(16)(22)6(30)(14)(19)(17)(20)(18) (21)(25)1(17)(23)(27)3(19)(22)(26)2(18)(24)(28)4(20)(31)7(23) (29)5(21)(32)8(24)(30)6(22)9(25)(13)(29)(11)(27)(15)(31)(10) (26)(14)(30)(12)(28)(16)(32)

Table 2: k-uniform words for cubes Q_k.

Example 5.12. Let us make a word for the Cartesian product $K_3 \square K_2$, see Figure 24. We know that K_3 can be 2-represented by the word 123123. So $K_3 \square K_2$ can be represented by the 3-uniform word $w' = 123415263456142536$, where all the nodes x_1 were renamed to x and all nodes x_2 were renamed to $x+3$. Figure 5.12 shows this graph and we see that this graph forms a 3-prism, Pr_3.

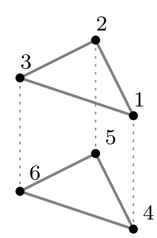


Figure 24: The 3-prism Pr_3.

We can use this result to formulate and prove the following corollary.

Corollary 5.13. The Cartesian product $K_n \square K_2$ has representation number n for $n \in \{1, 2\}$, and representation number 3 for all $n > 2$.

Proof. $K_1 \square K_2$ is equal to K_2, having representation number 1.

$K_2 \square K_2$ is the 4-cycle, which is known to have representation number 2, see Remark 4.56.

$K_3 \square K_2$ is equal to the 3-prism, for which we have shown in Corollary 5.11 that the representation number is 3.

If $n > 3$ then $K_n \square K_2$ contains the 3-prism as induced subgraph and thus cannot be 2-represented. Theorem 5.3 gives a 3-representation, because K_n has a 2-representation $12 \cdots n12 \cdots n$, so the representation number is 3. □

5.2 Cartesian product with K_n

The ideas used in Theorem 5.3 can be applied to formulate and prove the following generalisation.

The nodes of $G \,\Box\, K_n$ are denoted by $x_1, x_2, \ldots x_n$ for x running over the nodes of G; two nodes x_i, y_j are connected by an edge in $G \,\Box\, K_n$ if and only if

- $i = j$ and (x, y) is an edge in G, or
- $i \neq j$ and $x = y$.

Write V_i for the set of nodes x_i, so $V_1 \cup V_2 \cup \cdots \cup V_n$ is the set of nodes[2] of $G \,\Box\, K_n$.

Theorem 5.14. Let G be a k-representable graph for $k > 1$ and let w be a k-representant of G. Then the graph $G \,\Box\, K_n$ is $(k + (n-1))$-representable with representant $w' = f_n(w) f_{n-1}(w) \cdots f_1(w)$ for the occurrence based functions f_i defined by

$$f_1(x,i) = \begin{cases} x_1 & \text{if } i = 1 \\ x_n x_{n-1} \ldots x_1 & \text{if } 1 < i \leq k \end{cases}$$

and

$$f_j(x,i) = \begin{cases} x_j & \text{if } i = 1 \\ x_{j-1} \ldots x_1 x_n \ldots x_j & \text{if } i = 2 \\ \epsilon & \text{if } 2 < i \leq k \end{cases}$$

for $j = 2, \ldots, n$.

Proof. For every x the word $f_1(w)$ contains k copies of x_1 and $k-1$ copies of x_i for $i > 2$, and the words $f_j(w)$ contain 2 copies of x_j and 1 copy of x_i for $i \neq j$. So for every i, x_i occurs either $k + (n-1)$ times if $i = 1$, or $(k-1) + 2 + (n-2) = k + (n-1)$ times if $i \neq 1$. So w' is $(k + (n-1))$-uniform.

We have to prove that x_i, y_j alternate in w' for $x_i \neq y_j$ if and only if (x_i, y_j) is an edge in $G \,\Box\, K_n$, for $i, j = 1, 2, \ldots, n$, more precisely:

- if $x \neq y$ and $i = j$ then x_i, y_j alternate in w' if and only if x, y alternate in w,
- if $x = y$ and $i \neq j$ then x_i, y_j do alternate in w', and
- if $x \neq y$ and $i \neq j$ then x_i, y_j do not alternate in w'.

We do this by considering all cases separately. As the function f_j has a different form when $j = 1$, we will often make a case distinction based on this.

Let $x \neq y$ and $i = j = 1$. Observe that $f_1(w)_{V_1} = w_1$ and $f_l(w)_{V_1} = p_2(w_1)$ for all $l > 1$, in which w_1 is a copy of w where every symbol is indexed by 1. Now x_1, y_1 alternate in w' if and only if they alternate in $w'_{V_1} = f_n(w)_{V_1} f_{n-1}(w)_{V_1} \ldots f_1(w)_{V_1} = (p_2(w_1))^{n-1} w_1$, and by Lemma 3.23 for $A_1 = A_2 = \ldots = A_{n-1} = \{2\}$ and $A_n = N_k$, with $N_k = \{1, \ldots, k\}$, this holds if and only if x, y alternate in w, which we had to prove.

[2] Note that we could have taken the set of nodes as the disjoint union of n times V, but doing so would force us to talk about pairs of letters, which could be confusing as the labelling function, defined in Definition 3.20, uses the same notation.

Let $x \neq y$ and $i = j \geq 2$. Observe that $f_1(w)_{V_i} = p_{N_k \setminus \{1\}}(w_i)$ and $f_l(w)_{V_i} = p_{\{1,2\}}(w_i)$ for all $l > 1$, in which w_i is a copy of w in where symbol is indexed by i. Now x_i and y_i alternate in w' if and only if they alternate in $w'_{V_i} = f_n(w)_{V_i} f_{n-1}(w)_{V_2} \dots f_1(w)_{V_i} = (p_{\{1,2\}}(w_i))^{n-1} p_{N_k \setminus \{1\}}(w_i)$, and by Lemma 3.23 for $A_1 = A_2 = \dots = A_{n-1} = \{1, 2\}$ and $A_n = N_k \setminus \{1\}$ this holds if and only if x and y alternate in w, which we had to prove.

Let $x = y$, $i = 1$ and $i < j$. Then x_1, x_j alternate in w' since $w'_{\{x_1,x_j\}} = (x_j x_1)^{n-j}(x_j x_1 x_j)(x_1 x_j)^{j-2}(x_1(x_j x_1)^{k-1}) = (x_j x_1)^{k+(n-1)}$.

Let $x = y$ and without loss of generality $1 \neq i < j$. Then x_i, x_j alternate in w' since $w'_{\{x_i,x_j\}} = (x_j x_i)^{n-j}(x_j x_i x_j)(x_i x_j)^{j-i-1}(x_i x_j x_i)(x_j x_i)^{i-2}(x_j x_i)^{k-1} = (x_j x_i)^{k+(n-1)}$.

Now, let $x \neq y$ and $i < j$.

If $w_{\{x,y\}} = (xy)^k$ and $i = 1$, then $f_1(w)_{\{x_1,y_j\}}$ starts by $x_1 x_1$, so x_1, y_j do not alternate in w'.

If $w_{\{x,y\}} = (xy)^k$ and $i > 1$, then $f_i(w)_{\{x_i,y_j\}} = x_i x_i y_j$, so x_i, y_j do not alternate in w'.

If $w_{\{x,y\}} = (yx)^k$ then $f_j(w)_{\{x_i,y_j\}} = y_j y_j x_i$, so x_i, y_j do not alternate in w'.

In the remaining case x, y do not alternate in w, so $w_{\{x,y\}}$ contains either xx or yy. If it is xx, or it is yy and $w_{\{x,y\}}$ does not start in yy, then $f_1(w)_{\{x_1,y_j\}}$ contains $x_1 x_1$ for all $j > 1$ and $f_i(w)_{\{x_i,y_j\}} = x_i x_i y_j$ for all $1 \neq i < j$. Otherwise $w_{\{x,y\}}$ starts in yy, but then $f_j(w)_{\{x_i,y_j\}} = y_j y_j x_i$ for all $1 \neq i < j$. In all cases we conclude that x_i, y_j do not alternate in w', concluding the proof. □

Remark 5.15. This theorem does not always give the most optimal word possible. For instance, when we make a word for $K_2 \,\square\, K_3$ using Theorem 5.14 we will find a $2+(3-1) = 4$ representation for the 3-prism, while $K_3 \,\square\, K_2$ gives a $2 + (2 - 1) = 3$ representation for the same graph, as we saw in Example 5.12.

In Chapter 6 we will formulate some questions regarding optimality.

Remark 5.16. We must note that Theorem 5.14 does allow the use of some decomposition, i.e. a word for the graph $K_2 \square K_3 \square K_4 \square K_7$ could be obtained by applying this construction successively. Again, this does not guaranteed to obtain the shortest word possible, but the word can be kept shorter by choosing the order in the construction. For example $((K_7 \,\square\, K_3) \,\square\, K_4) \,\square\, K_2$ results in a $((2 + 2) + 3) + 1 = 8$-uniform word, while $((K_2 \,\square\, K_4) \,\square\, K_3) \,\square\, K_7$ results in a $((2 + 3) + 2) + 6 = 13$-uniform word.

Example 5.17. We will now construct a word for the product $G \,\square\, K_3$ with G the graph in Figure 22a, represented by the word $w = 3142132545$. The result of this product is the graph shown in Figure 25. Following the construction in Theorem 5.14 we find:

- $f_1(w) = 3_1 1_1 4_1 2_1 1_3 1_2 1_1 3_3 3_2 3_1 2_3 2_2 2_1 5_1 4_3 4_2 4_1 5_3 5_2 5_1$;
- $f_2(w) = 3_2 1_2 4_2 2_2 1_1 1_3 1_2 3_1 3_3 3_2 2_1 2_3 2_2 5_2 4_1 4_3 4_2 5_1 5_3 5_2$;
- $f_3(w) = 3_3 1_3 4_3 2_3 1_2 1_2 1_3 3_2 3_1 3_3 2_2 2_1 2_3 5_2 4_2 4_1 4_3 5_2 5_1 5_3$.

Now, renaming every x_i to $x + (i - 1) \cdot 5$ and concatenating these words we get:
$w' = (13)(11)(14)(12)66(11)83(13)72(12)(10)94(14)(10)5(15)86971(11)63(13)82(12)7(10)\dots$
$4(14)95(15)(10)3142(11)61(13)83(12)725(14)94(15)(10)5.$

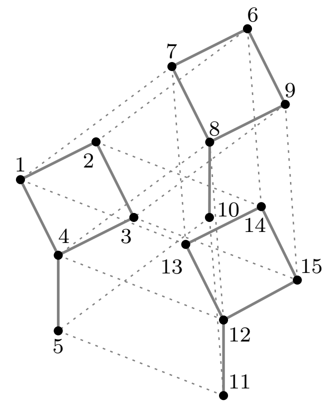


Figure 25: Cartesian product of G in Figure 22a and K_3.

5.3 Cartesian product with a permutationally representable graph

Until now we have focused on the Cartesian product with a complete graph, in which we use the fact that the complete graph can be permutationally represented by one permutation. We will now generalise Theorem 5.14 to form a word for the Cartesian product of a graph $G = (V_G, E_G)$ with any permutationally representable graph $H = (V_H, E_H)$, so a comparability graph.

As before, the nodes of $G \,\square\, H$ are denoted by $x_1, x_2, \dots x_{\#V_H}$ for all $x \in V_G$ G and the nodes of H renamed to 1 up to $\#V_H$; two nodes x_i, y_j are connected by an edge in $G \,\square\, H$ if and only if

- $i = j$ and (x, y) is an edge in G, or
- $x = y$ and (i, j) is an edge in H.

Again we write V_i for the set of nodes x_i, so $V_1 \cup V_2 \cup \dots \cup V_{\#V_H}$ is the set of nodes of $G \,\square\, H$.

Theorem 5.18. Let $G = (V_G, E_G)$ be a k-representable graph for $k > 1$ and let w be a k-representant of G. Also, let $H = (V_H, E_H)$ be l-permutationally represented by $v = p_1 p_2 \dots p_l$, consisting of l permutations p_i of the letters $\{1, 2, \dots, n\}$. Then the graph $G \,\square\, H$ is $l \cdot (k + (n-1))$-representable with representant
$w' = g_1(v')g_2(v') \dots g_l(v')$, where $v' = f_n(w)f_{n-1}(w) \dots f_1(w)$, and the occurrence based functions f_i, g_i defined by

$$f_1(x, i) = \begin{cases} x_1 & \text{if } i = 1 \\ x_n x_{n-1} \dots x_1 & \text{if } 1 < i \leq k \end{cases},$$

$$f_j(x, i) = \begin{cases} x_j & \text{if } i = 1 \\ x_{j-1} \dots x_1 x_n \dots x_j & \text{if } i = 2 \\ \epsilon & \text{if } 2 < i \leq k \end{cases}$$

for $j = 2, \dots, n$ and

$$g_j(x_i, t) = x_{p_j(i)}$$

for all t and $j = 1, \dots, l$, where $p_j(i)$ is the ith character of the jth permutation of the word v.

Proof. For every x the word $f_1(w)$ contains k copies of x_1 and $k-1$ copies of x_i for $i > 2$, and the words $f_j(w)$ contain 2 copies of x_j and 1 copy of x_i for $i \neq j$. So for every i, x_i occurs either $k + (n-1)$ times if $i = 1$, or $(k-1) + 2 + (n-2) = k + (n-1)$ times if $i \neq 1$. So v' is $(k + (n-1))$-uniform. As each p_j is a permutation and contains every letter exactly once, applying g_j on a uniform word keeps the word uniform. So w' is $l \cdot (k + (n-1))$-uniform.

We have to prove that x_i, y_j alternate in w' for $x_i \neq y_j$ if and only if (x_i, y_j) is an edge in $G \square H$, for $i, j = 1, 2, \ldots, n$, more precisely:

- if $x \neq y$ and $i = j$ then x_i, y_j alternate in w' if and only if x, y alternate in w,
- if $x = y$ and $i \neq j$ then x_i, y_j alternate in w' if and only if i, j alternate in v, and
- if $x \neq y$ and $i \neq j$ then x_i, y_j do not alternate in w'.

We do this by considering all cases separately. As the function f_j has a different form when $j = 1$, we will often make a case distinction based on this.

Let $x \neq y$ and $i = j$. The function g does not change the letters, it only changes the index of the letters. So x and y alternate in w' if and only if they do in v', which we will now prove.

Let $i = j = 1$. Observe that $f_1(w)_{V_1} = w_1$ and $f_i(w)_{V_1} = p_2(w_1)$ for all $i > 1$, in which w_1 is a copy of w where every symbol is indexed by 1. Now x_1 and y_1 alternate in v' if and only if they alternate in $v'_{V_1} = f_n(w)_{V_1} f_{n-1}(w)_{V_1} \ldots f_1(w)_{V_1} = (p_2(w_1))^{n-1} w_1$, and by Lemma 3.23 for $A_1 = A_2 = \ldots = A_{n-1} = \{2\}$ and $A_n = N_k$, with $N_k = \{1, \ldots, k\}$, this holds if and only if x and y alternate in w, which we had to prove.

Let $i = j \geq 2$. Observe that $f_1(w)_{V_i} = p_{N_k \setminus \{1\}}(w_i)$ and $f_l(w)_{V_i} = p_{\{1,2\}}(w_i)$ for all $l > 1$, in which w_i is a copy of w where every symbol is indexed by i. Now x_i, y_i alternate in v' if and only if they alternate in $v'_{V_i} = f_n(w)_{V_i} f_{n-1}(w)_{V_2} \ldots f_1(w)_{V_i} = (p_{\{1,2\}}(w_i))^{n-1} p_{N_k \setminus \{1\}}(w_i)$, and by Lemma 3.23 for $A_1 = A_2 = \ldots = A_{n-1} = \{1, 2\}$ and $A_n = N_k \setminus \{1\}$ this holds if and only if x, y alternate in w, which we had to prove.

Let $x = y$ and $i \neq j$. In this case we first prove that x_i, x_j alternate in v' for all i and j and then use the functions g_t to prove that x_i and x_j alternate in w' if and only if i, j alternate in v.

Let $x = y$ and without loss of generality $i = 1$ and $i < j$. Then x_1 and x_j alternate in v' since
$v'_{\{x_1, x_j\}} = (x_j x_1)^{n-j} (x_j x_1 x_j)(x_1 x_j)^{j-2} (x_1 (x_j x_1)^{k-1}) = (x_j x_1)^{k+(n-1)}$.

Let $x = y$ and without loss of generality $1 \neq i < j$. Then x_i, x_j alternate in v' since
$v'_{\{x_i, x_j\}} = (x_j x_i)^{n-j} (x_j x_i x_j)(x_i x_j)^{j-i-1} (x_i x_j x_i)(x_j x_i)^{i-2} (x_j x_i)^{k-1} = (x_j x_i)^{k+(n-1)}$.

The functions g_j only change the order in which x_i and x_j occur in the resulting word. Also note that if x_i and x_j alternate in v', then for all s the letters x_i and x_j alternate in $g_s(v')$, and
$g_s(v')_{\{x_i, x_j\}} = x_i x_j \ldots$ if and only if $(p_s)_{\{i,j\}} = ij$. We see that if i, j alternate in v, then the order of x_i and x_j is changed in the same way in each g_t, and thus x_i, x_j alternate in w'. Now if i, j do not alternate in v, then there must be an s and t such that $(p_s)_{\{x_i, x_j\}} = x_i x_j$ and $(p_t)_{\{x_i, x_j\}} = x_j x_i$, and thus $g_s(v')_{\{x_i, x_j\}} = x_i x_j \ldots$ and $g_t(v')_{\{x_i, x_j\}} = x_j x_i \ldots$ and x_i, x_j do not alternate in w'.

Now, let $x \neq y$ and, without loss of generality, $i < j$.

If $w_{\{x,y\}} = (xy)^k$ and $i = 1$, then $f_1(w)_{\{x_1,y_j\}}$ starts by x_1x_1, so $x_{p_t(1)}, y_{p_t(j)}$ do not alternate in v' for all t.

If $w_{\{x,y\}} = (xy)^k$ and $i > 1$, then $f_i(w)_{\{x_i,y_j\}} = x_ix_iy_j$, so $x_{p_t(i)}, y_{p_t(j)}$ do not alternate in v' for all t.

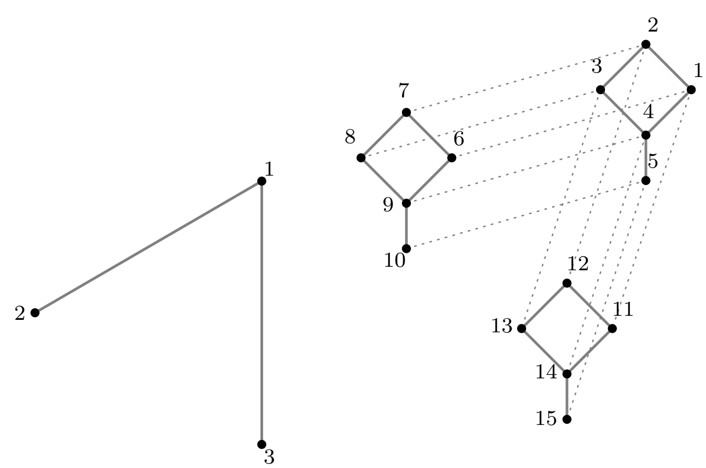
If $w_{\{x,y\}} = (yx)^k$ then $f_j(w)_{\{x_i,y_j\}} = y_jy_jx_i$, so $x_{p_t(i)}, y_{p_t(j)}$ do not alternate in v' for all t.

In the remaining case x and y do not alternate in w, so $w_{\{x,y\}}$ contains either xx or yy. If it is xx, or it is yy and $w_{\{x,y\}}$ does not start in yy, then $f_1(w)_{\{x_1,y_j\}}$ contains x_1x_1 for all $j > 1$ and $f_i(w)_{\{x_i,y_j\}} = x_ix_iy_j$ for all $1 \neq i < j$. Otherwise $w_{\{x,y\}}$ starts in yy, but then $f_j(w)_{\{x_i,y_j\}} = y_jy_jx_i$ for all $1 \neq i < j$. In all cases we conclude that $x_{p_t(i)}, y_{p_t(j)}$ do not alternate in v' for all t, concluding the proof. □

Remark 5.19. It is a known fact that $G \square H$ is a comparability graph if and only if both G and H are bipartite, see[3]. In the case where both G and H are bipartite Theorem 5.18 does not yield a permutation-representation for $G \square H$.

Remark 5.20. In Remark 4.25 we noted that the permutation-representation number of a graph can be lower than the actual representation number. Theorem 5.18 uses a permutation-representation of H, so it might seem logical that, in the case where the permutation-representation number of H is lower than its representation number, this construction is not ideal. We will discuss this question, and some others regarding optimality in Chapter 6.

Example 5.21. We will once again use the graph in Figure 22a as G. Figure 26a shows the graph we will use for H, which can be permutationally represented by $v = 123132$.



(a) The graph H.   (b) $G \square H$.

Figure 26: Example of Cartesian product with a permutationally representable graph.

Using Theorem 5.18 we get the following:

- $w = 3142132545$, $k = 2$;
- $v = 123132$, $l = 2$, $n = 3$, $p_1 = 123$, $p_2 = 132$;

- $f_1(w) = 3_1 1_1 4_1 2_1 1_3 1_2 1_1 3_3 3_2 3_1 2_3 2_2 2_1 5_1 4_3 4_2 4_1 5_3 5_2 5_1$
  $f_2(w) = 3_2 1_2 4_2 2_2 1_1 1_3 1_2 3_1 3_3 3_2 2_1 2_3 2_2 5_2 4_1 4_3 4_2 5_1 5_3 5_2$
  $f_3(w) = 3_3 1_3 4_3 2_3 1_2 1_1 1_3 3_2 3_1 3_3 2_2 2_1 2_3 5_3 4_2 4_1 4_3 5_2 5_1 5_3$
  $v' = f_3(w) f_2(w) f_1(w)$

- $g_1(f_1(w)) = 3_1 1_1 4_1 2_1 1_3 1_2 1_1 3_3 3_2 3_1 2_3 2_2 2_1 5_1 4_3 4_2 4_1 5_3 5_2 5_1$
  $g_2(f_1(w)) = 3_1 1_1 4_1 2_1 1_2 1_3 1_1 3_2 3_3 3_1 2_2 2_3 2_1 5_1 4_2 4_3 4_1 5_2 5_3 5_1$
  $g_1(f_2(w)) = 3_2 1_2 4_2 2_2 1_1 1_3 1_2 3_1 3_3 3_2 2_1 2_3 2_2 5_2 4_1 4_3 4_2 5_1 5_3 5_2$
  $g_2(f_2(w)) = 3_3 1_3 4_3 2_3 1_1 1_2 1_3 3_1 3_2 3_3 2_1 2_2 2_3 5_3 4_1 4_2 4_3 5_1 5_2 5_3$
  $g_1(f_3(w)) = 3_3 1_3 4_3 2_3 1_2 1_1 1_3 3_2 3_1 3_3 2_2 2_1 2_3 5_3 4_2 4_1 4_3 5_2 5_1 5_3$
  $g_2(f_3(w)) = 3_2 1_2 4_2 2_2 1_3 1_1 1_2 3_3 3_1 3_2 2_3 2_1 2_2 5_2 4_3 4_1 4_2 5_3 5_1 5_2.$

Combining these words and renaming x_i to $x + (i-1) \cdot 5$ gives the word:
$w' = 8697(11)16(13)38(12)27(10)(14)49(15)5(10)(13)(11)(14)(12)16(11)38(13)27\ldots$
$(12)(15)49(14)5(10)(15)31426(11)18(13)37(12)259(14)4(10)(15)5(13)(11)(14)(12)\ldots$
$61(11)83(13)72(12)(15)94(14)(10)5(15)86971(11)63(13)82(12)7(10)4(14)95(15)(10)\ldots$
$3142(11)61(13)83(12)725(14)94(15)(10)5$, and the graph is shown in Figure 26b.

6 Open problems

The focus of this chapter is to discuss some open problems that arise from the new found results in Sections 4.4 and Chapter 5.

6.1 Bipartite graphs

As mentioned in Section 4.4, at this point in time it is not clear if this result is new. The permutation-representation number of a comparability graph is the same as the dimension of the poset, see Definition 4.21.

There are many results on the dimension of posets, see for instance [10], but none have been found that give a lower upper bound for the specific setting discussed here. We can say that the discussed construction often gives the smallest possible permutation-representation. As we have seen in Example 4.26, the cube Q_3 has permutation-representation number 4, which also results from Theorem 4.45. However, we have not been able to find a bipartite graph that has a lower permutation-representation number than found with Theorem 4.45. This gives rise to the following question.

Question 6.1. Is the bound on the representation in Proposition 4.44 sharp?

Other questions that are related to this are, for example, the following.

Question 6.2. Is the permutation-representation number of the k-cube equal to 2^{k-1}?

Question 6.3. Is the permutation-representation number of the cycle graph C_n equal to $\frac{n}{2}$?

Linking this result to the dimension of posets, the following question arises.

Question 6.4. What are the requirements for a poset to have its comparability graph be permutationally-representable using Proposition 4.44 and what are the known implications of those requirements on the dimension of this poset?

It is not clear right now if this family of posets intersects with any known families of posets. It might be interesting for future research to look deeper into the connection between posets with bipartite comparability graphs and their word-representations.

6.2 Cartesian products

As we have seen in Remark 5.15, the construction of Theorem 5.14, and with that Theorem 5.3, is not always optimal. However, we did find an optimal representation using Theorem 5.14 in Corollary 5.13. The next example will illustrate that, even if the graph we are concerned with is built up as the Cartesian product of several complete graphs, we do not obtain the shortest word possible.

Example 6.5. We will look at the Cartesian product $(K_3 \square K_2) \square K_2$, see Figure 27, where $K_3 \square K_2$ is the 3-prism we saw in Example 5.12.

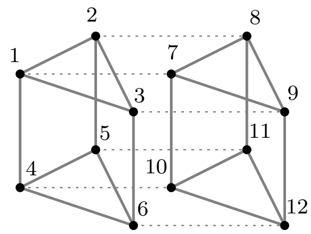


Figure 27: Cartesian product of Pr_3 and K_2, 3-represented by
$w = 183(10)75(12)9(11)4(10)6(12)285(11)17324165(10)94(12)3682(11)79$.

Following Theorem 5.3 we obtain a 4-uniform word, however, this graph can also be 3-represented by the word[3] given in the caption. We also know that this graph contains the 3-prism as induced subgraph, therefore it is not 2-representable, so the representation number is 3.

We know from Corollary 5.6 that if G has representation number k, then $G \,\square\, K_2$ has representation number k or $k+1$. Therefore we can ask the following question.

Question 6.6. Give a characterisation for the graphs G with representation number k such that $G \,\square\, K_2$ also has representation number k.

We also noted in Theorem 5.8 that the k-cube is k-representable. It is known that when $k \leq 4$, the k-cube has representation number k, but we do not know if this holds for larger k.

Question 6.7. Is there a k such that the k-cube is l-representable with $l < k$?

Following this and Question 6.2 we took a look at the relation between the permutation-representation number and the representation number of a comparability graph. We know that the permutation-representation number will be greater than or equal to the representation number, but as we have seen before in Example 4.26 they are not necessarily equal. However, we suspect that there is a characterisation of graphs for which these two are equal.

Question 6.8. Characterise the graphs for which the representation number equals the permutation-representation number.

Furthermore, as noted in Remark 5.19, we know that the resulting graph $G \,\square\, H$ from Theorem 5.18 is permutationally representable if and only if G and H are both bipartite. As the Cartesian product of two bipartite graphs is bipartite itself, it would help gaining insight in the problem of finding (permutation-)representations by looking at the words obtained from both Theorem 5.18 and Theorem 4.45 to find some structure.

Now when we focus on the main question we have regarding Theorems 5.3, 5.14 and 5.18, we can formulate it as follows.

Question 6.9. Specify necessary and sufficient conditions on G and H such that the construction in Theorem 5.18 results in the shortest word for the resulting Cartesian product.

[3]This word was obtained by using a tool by Hans Zantema, which can be downloaded from `http://www.win.tue.nl/ hzantema/reprnr.html`.

In the light of Remarks 5.15 and 5.16 there is a possibility that the answer to this question is related to the decomposition of a graph as Cartesian product of other graphs. We might expect a certain notion of minimal representing graphs, of which a large number of graphs can be constructed via Cartesian products and constructions as in Chapter 4. Possibly good candidates to start with are graphs with low representation number, such as the complete graphs and cycle graphs.

Chapter 5 shows two progressive steps in generalising an algorithm to accommodate a wider variety of graphs. As we know from Theorem 5.1, for all G and H that are word-representable, $G \square H$ is word-representable, but we have not been able to find an algorithm that works for arbitrary H. It is not clear if Theorem 5.18 can be generalised or that this needs to be done in another way. From this the following question arises.

Question 6.10. Can Theorem 5.18 be generalised to produce a word for the Cartesian product $G \square H$ with G and H arbitrary word-representable graphs?

References

[1] Ö. Akgün, I. Gent, S. Kitaev, H. Zantema. Solving computational problems in the theory of word-representable graphs. Available at `http://arxiv.org/abs/1808.01215`, 2018.

[2] J. Bondy, U. Murty. *Graph Theory*. Springer London, 2011.

[3] A. Bretto. Comparability Graphs and Digital Topology. *Computer Vision and Image Understanding*, **82**:33–41, 2001.

[4] B. Broere, H. Zantema. The k-cube is k-representable. Available at `http://arxiv.org/abs/1808.01800`, 2018.

[5] A. Gao, S. Kitaev, P. Zhang. On 132-representable graphs. *Australasian Journal of Combinatorics*, **69**(1):105–118, 2017.

[6] M. Glen, S. Kitaev, A. Pyatkin. On the representation number of a crown graph. *Discrete Applied Mathematics*, **244**:89–93, 2018.

[7] M. Halldórson, S. Kitaev, A. Pyatkin. The NP-completeness of edge-coloring. *Discrete Applied Mathematics*, **201**:164–171, 2016.

[8] I. Holyer. Semi-transitive orientations and word-representable graphs. *SIAM Journal on Computing*, **10**:718–720, 1981.

[9] M. Jones, S. Kitaev, A. Pyatkin, J. Remmel. Representing graphs via pattern avoiding words. *Electronic Journal of Combinatorics*, **22**(2):20–53, 2015.

[10] G. Joret, P. Micek, W. Trotter, R. Wang, V. Wiechert. On the Dimension of Posets with Cover Graphs of Treewidth 2. *Order*, **34**(2):185–234, 2017.

[11] S. Kitaev. On graphs with representation number 3. *Journal of Automata, Languages and Combinatorics*, **18**:97–112, 2013.

[12] S. Kitaev, V. Lozin. *Words and graphs*. Springer Berlin Heidelberg, 2015.

[13] S. Kitaev, A. Pyatkin. On representable graphs. *Journal of Automata, Languages and Combinatorics*, **13**(1):45–54, 2008.

[14] S. Kitaev, S. Seif. Word Problem of the Perkins Semigroup via Directed Acyclic Graphs. *Order*, **25**(3):177–194, 2008.

[15] Y. Mandelshtam. On graphs representable by pattern-avoiding words. Available at `http://arxiv.org/abs/1608.07614`, 2016.

[16] W. Trotter. *Combinatorics and partially ordered sets: Dimension theory*. The Johns Hopkins University Press, Baltimore, 1992.

[17] R. Wilson. *Four colors suffice*. Princeton University Press, Princeton, NJ, 2014.

[18] M. Yannakakis. The complexity of the partial order dimension problem. *SIAM Journal on Algebraic and Discrete Methods*, **3**:351–358, 1982.