Introduction to Graphs Minors, Tree Decompositions and FPT Algorithms.

MPRI
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4</td>
<td>Computing tree width</td>
<td>45</td>
</tr>
<tr>
<td>5.5</td>
<td>FPT Algorithms</td>
<td>46</td>
</tr>
<tr>
<td>5.5.1</td>
<td>Existence of FPT via Robertson and Seymour Theorems</td>
<td>47</td>
</tr>
<tr>
<td>5.5.2</td>
<td>TreeWidth Based Algorithms</td>
<td>47</td>
</tr>
<tr>
<td>5.5.3</td>
<td>Branching</td>
<td>47</td>
</tr>
<tr>
<td>5.5.4</td>
<td>Kernelization</td>
<td>48</td>
</tr>
<tr>
<td>5.5.5</td>
<td>Iterative Compression</td>
<td>48</td>
</tr>
<tr>
<td>5.5.6</td>
<td>Colour Coding (NON TRAITÉ CETTE ANNEE)</td>
<td>49</td>
</tr>
<tr>
<td>5.6</td>
<td>Exercises</td>
<td>49</td>
</tr>
</tbody>
</table>
Introduction

These are notes in complement to a 12h lecture given in 2016-2017 in the MPRI (Master Parisien de Recherche en Informatique). The objective of the course is to serve as an introduction to some of the concepts used in the proof of Wagner’s Conjecture given by Robertson and Seymour, and to show some algorithmic consequences of their work, in particular with regards to FPT Algorithms.

For a general textbook on graph theory, the book of Bondy and Murty ([7] is an excellent reading, but the theory of graphs minors and tree decomposition is not deeply discussed. On this particular topic, there exists a good a short introduction by Lovasz ([17]), but the main influence for this course is Chapter 12 of the book of Diestel ([10]), which is also a good source for exercises. For Fixed Parameter Tractable Algorithms, a classic book is ([11]).
Chapter 1

Introduction

1.1 Basic Definitions and Terminology

In this course, a graph is given by a set \( V \), whose elements are called vertices, and a set \( E \) whose elements, called edges of the graphs, are distinct subsets of size 2 of \( V \). According to the usual vocabulary, this means that our graph will always be simple and without loops. Unless specified, \( V \) will always be a finite set. For a graph \( G \), \( V(G) \) will always denote its set of vertices, \( E(G) \) its set of edges. Very often we will write \( xy \) instead of \( \{x, y\} \) for an edge of \( G \).

A vertex \( v \) is a neighbour of a vertex \( u \) if \( uv \in E(G) \). The neighbourhood of \( u \), denoted \( N(u) \) is the set of neighbours of \( u \). Its degree, denoted \( d(u) \) is the cardinality of its neighbourhood. The maximum degree of a graph is usually denoted \( \Delta \). A graph with no edges will be called a stable set, or independent set, and a graph will all possible edges between its vertices a clique, or complete graph. The complete graph on \( n \) vertices is usually denoted \( K_n \).

The path \( P_k \) is a graph with \( V(P_k) = \{x_1, x_2, \ldots, x_k\} \), with edges \( E = \{x_i, x_{i+1}, \ 1 \leq i \leq k - 1\} \). The vertices \( x_1 \) and \( x_k \) are called the endpoints of the path. If we add the edge \( x_kx_1 \) to \( P_k \) then the resulting graph is the circuit on \( k \) vertices, denoted \( C_k \).

1.2 Three Algorithmic Problems

Consider the following problem of connectivity.

Problem : \( k \) disjoint path problem
Input : A graph \( G \), an integer \( k \) and two subsets of vertices \( A \) and \( B \) of size \( k \)
Output : TRUE if there exists \( k \) vertex disjoint paths from \( A \) to \( B \)?

This problem is a very classical one, and Ford-Fulkerson Algorithm tells us that this is solvable in time \( O((k|E(G)|)) \) (classical Ford-Fulkerson Algorithm is for edge disjoint path in the directed case, but it is easy to reduce our case to this one). The maximum value \( k \) corresponds to a minimum vertex cut separating \( A \) and \( B \) and is a classical result of Menger.

Theorem 1.1 (Menger,1927,[18])

Let \( x \) and \( y \) be distinct vertices of a graph \( G \). Then the minimum number of vertices whose deletion separates \( x \) from \( y \) is equal to the maximum number of internally disjoint paths between \( x \) and \( y \).

Proof. See [7].
Problem: $k$-disjoint rooted path problem
Input: A graph $G$, an integer $k$, and two subsets of vertices $X = \{x_1, x_2, \ldots, x_k\}$ and $Y = \{y_1, y_2, \ldots, y_k\}$
Output: TRUE if there exists disjoint paths $P_1, P_2, \ldots, P_k$, such that $P_i$ is a path from $x_i$ to $y_i$.

This kind of problem in a more general form is known as commodity flow problem and has many applications. With $k$ part of the input, this problem is NP-complete, even restricted to the class of planar graphs. Nevertheless, in the Graph Minor series of papers, Robertson and Seymour proved a polynomial algorithm for fixed $k$. This result is extremely difficult and relies on techniques and notions that will be illustrated in this course.

Theorem 1.2 (Robertson-Seymour, [19])

The $k$-disjoint path problem can be solved in time $O(f(k) n^3)$

The result has been improved to quadratic time by Kawabayarashi, Kobashi and Reed ([]). Let us see an algorithmic consequence of this result related to topological minor detection.

Definition 1.3

A graph $H$ is topological minor of a graph $G$ if there exists an injective mapping $f$ from $V(H)$ to $V(G)$ such that for each edge $uv$ of $H$, there exists in $G$ a path $P_{uv}$ connecting $f(u)$ and $f(v)$ in $G$ with the property that all these paths are internally disjoint.

Example. Describe the graphs that do not contain the following graphs as topological minors: $K_3$, $K_{1,3}$, $K_{1,4}$.

A natural algorithmic problem is then the following.

Problem: Topological $H$-minor detection
Input: A graph $G$ and a graph $H$.
Output: TRUE if $H$ is a topological minor of $G$, FALSE otherwise.

The problem is NP-complete if $H$ is part of the input, but if $H$ but if $H$ is fixed, then this problem was proven to be polynomial by Robertson and Seymour.

Theorem 1.4

Let $H$ be a fixed graph. There exists a polynomial time algorithm to decide whether $H$ is a topological minor of a given graph $G$.

Proof. Let $f : V(H) \rightarrow V(G)$ be an injection (note there are polynomially many such objects), we want to decide if there exists disjoint paths in $G$ between the $f(v)$ corresponding to edges of $H$. To do that, we replace each vertex $f(v)$ by $d_H(v)$ copies of $f(v)$ (having the same neighbours). Now, for $k = |E(H)|$, solving the $k$-rooted disjoint paths problem for these sources clearly solves the desired question. \hfill \square

The complexity of this algorithm is hence $O(f(k)) n^k$, where $k$ is the size of $H$, and $n$ the size of $G$. It is therefore polynomial for every fixed $k$. In 2010, Grohe, Kawabayarashi, Marx, and Wollan proved a stronger result, that this can be done in $O(f(k)) n^3$. Such an algorithm is called
Fixed Parameter Tractable (FPT) algorithm. We will discuss more about those in the last chapter of this course.

In particular, the previous theorem implies that any family of graphs that is defined with forbidding a FINITE family of graphs as topological minors is polynomially testable. One such family is very well known, it is the family of planar graphs, as was proven by Kuratowski in 1930.

**Theorem 1.5 (Kuratowski, 1930)**

| A graph $G$ is planar if and only if it does not contain $K_5$ or $K_{3,3}$ as a topological minor. |

Planar graph will play a central role in this course, and as we will see, this theorem will be characteristic of the kind of result we will be interested. The crucial fact here is not that planar graphs are defined by a certain list of forbidden topological minors, this is easy (why?), it is the finiteness of this list that is non trivial. The central result of Robertson and Seymour theory is that many different graph properties can be characterised by a finite list of forbidden substructures, and hence get polynomial time recognition algorithm. Of course, one does not need the difficult of Robertson and Seymour to prove that planar graphs are polynomially recognisable, there even exists linear time algorithm to do that. Nevertheless, we will see later that there are instances of such recognition problem for which the only proof of polynomiality was obtained through their results.

### 1.3 Minors

We define three operations on a graph $G$ (at the end of each line the notation for the resulting graph).

1. **Remove a vertex** $v$ (and all its incident edges) : $G \setminus v$
2. **Remove an edge** $e$ (but not its end vertices) : $G \setminus e$
3. **Contract an edge** $e = xy$, which means remove $x$ and $y$, add a new vertex $z$ whose neighbourhood is the union of the neighbourhoods of $x$ and $y$ (without putting any loop on $z$) : $G/e$.

A contraction $G/e$ is **topological** if one of the endpoints of $e$ has degree 2. Its inverse is the subdivision operation which consists in removing an edge $xy$, adding a new vertex $z$, and adding the edges $xz$ and $zy$.

**Definition 1.6**

<table>
<thead>
<tr>
<th>Let $G$ and $H$ be two graphs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>- $H$ is an <strong>induced subgraph</strong> of $G$ if $H$ is obtained from $G$ by the repeated use of rule 1.</td>
</tr>
<tr>
<td>- $H$ is a <strong>subgraph</strong> of $G$ if $H$ is obtained from $G$ by the repeated use of rule 1 and 2.</td>
</tr>
<tr>
<td>- $H$ is a <strong>spanning subgraph</strong> of $G$ if $H$ is obtained from $G$ by the repeated use of rule 2.</td>
</tr>
<tr>
<td>- $H$ is a <strong>minor</strong> of $G$ if $H$ is obtained from $G$ by the repeated use of rule 1, 2 and 3.</td>
</tr>
<tr>
<td>- $H$ is a <strong>topological minor</strong> of $G$ is $H$ is a minor of $G$ and every contraction used was topological.</td>
</tr>
</tbody>
</table>
Recall that the largest integer $k$ such that $G$ has a complete graph (resp. independent set) on $k$ vertices as an (induced) subgraph, is called the **clique number** (resp. **independence number**) of $G$, denoted $\omega(G)$ (resp. $\alpha(G)$). Due to a classical result of Karp ([15]), deciding if a graph has $\omega(G) \geq k$ or not (similarly for $\alpha(G)$) is an NP-hard problem.

The following lemma gives an alternate definition of minor that is often useful.

**Lemma 1.7**

Let $G$ and $H$ be two graphs, and denote $V(H) = \{v_1, \ldots, v_p\}$. Then $H$ is a minor of $G$ if and only if there exists $p$ connected and disjoint subgraphs $G_1, \ldots, G_p$ of $G$ such that for every edge $v_iv_j$ of $H$, there exists an edge between $G_i$ and $G_j$.

Let us mention here that high density implies the existence of a large minor; there exists theorems with better bounds, but we are only interested here in the fact that such a bounds exists.

**Theorem 1.8**

Every graph with average degree at least $2^{r-2}$ contains $K_r$ as a minor.

**Proof.** By induction on $r$. Let $G$ be a graph of average degree at least $2^{r-2}$. Therefore $|E(G)|/|V(G)| \geq 2^{r-3}$. Let $H$ be minimal amongst all minors of $G$ such that $|E(H)|/|V(H)| \geq 2^{r-3}$. It implies that when one contracts an edge in $H$, one must loose at least $2^{r-3}$ edges (otherwise the inequality would still be satisfied, and $H$ would not be minor minimal). Hence, for any $xy$ edge of $H$, $x$ and $y$ have at least $2^{r-3}$ common neighbours. In otherwords, if $x$ is a vertex in $H$, then the minimum degree in its neighbourhood is at least $2^{r-3}$, so by induction it contains a $K_{r-1}$ minor, which yields with $x$ the desired $K_r$ minor. \(\square\)

Let us discuss now the difference between minors and topological minors. Topological minors are special kind of minors but of course the converse is not true: a graph $G$ can contain $H$ as a minor, but not as a topological minor. (Exercise 1.5).

When $H$ is of small maximum degree, this is nevertheless true.

**Theorem 1.9**

Let $H$ be a graph with maximum degree at most 3. Then a graph $G$ has an $H$-minor if and only if it contains an $H$-subdivision.

**Proof.** Let $H$ be a graph with vertex set $V(H) = \{v_1, \ldots, v_p\}$ and assume it is a minor of $G$. We use Lemma 1.7: there exists $p$ connected and disjoint subgraphs $G_1, \ldots, G_p$ of $G$ such that for every edge $v_iv_j$ of $H$, there exists an edge between $G_i$ and $G_j$.

Any topological minor of the graph induced by the $G_i$ will be topological minor of $G$ so we apply topological minor operations. First between the $G_i$ we delete all edges but one between each pair when it is needed, that is when there was one in $H$ between the corresponding vertices. Then we can delete edges inside each $G_i$ keeping it connected and therefore assume that every subgraph $G_i$ is a tree. Also, every leaf of such a tree must be a vertex incident to an edge going to another $G_i$, otherwise it is not needed. Finally we can contract edges to get rid of degree 2 vertices. It is easy to see that what we get is then a single vertex in each $G_i$ and therefore the graph $H$ as a topological minor of $G$. \(\square\)

In the general case, using the same argument we are able to still prove a result that reduces minor detection to topological minor detection.
Theorem 1.10

For every graph $H$, there is a finite family $\mathcal{H}$ of graphs with the property that $G$ contains $H$ as a minor if and only if it contains some graph in $\mathcal{H}$ as a topological minor.

Proof. We start the proof exactly as in the previous result, and by again choosing minimal $G_i$, we now get for each $G_i$ a tree with at most $|H|$ leaves and no vertex of degree 2. There finitely many such trees (why?). So by replacing the vertices of $H$ by these trees in all possible ways, we obtain a finite collection of graphs $\mathcal{H}$ with the desired properties. □

This result combined with Theorem 1.4 now clearly implies the following theorem due to Robertson and Seymour (Graph Minors XIII [21])

Theorem 1.11 (Robertson and Seymour, 1995)

Let $H$ be a fixed graph. There exists a polynomial time algorithm to decide whether $H$ is a minor of a given graph $G$.

Of course this extends to the recognition of any family of graphs that is defined by a finite list of forbidden minors. The question is then to understand what are the families of graphs of this type. A trivial fact is that such families are closed under minors (every minor of a graph in the family is in the family). The celebrated Robertson and Seymour Theorem, solving a conjecture of Wagner from 1937 says that this is sufficient.

Theorem 1.12 (Robertson and Seymour)

A minor closed class of graph is defined by a finite list of forbidden minors

The proof of this conjecture is due to Robertson and Seymour, and is the main result of a series of more than 20 papers called the Graph Minors series, written in the 1980’s. The conjecture of Wagner is eventually proven in Graph Minors XX ([22]), published in 2004.

One of the most important special case of the conjecture is the generalisation of Kuratowski’s Theorem for any fixed surface. It was first proven by Archdeacon and Huneke for non-orientable surfaces ([4]), and for the general case by Robertson and Seymour in Graph Minor VIII [20].

Theorem 1.13

For every closed compact surface, the set of graphs that are embeddable in this surface has a finite set of bounds.

Another example is the question of linklessly embeddable graphs (graphs that can be embedded in three-dimensional spaces, so that two cycles can always be pulled apart, that is they don’t entwine like links in a chain). Before Robertson and Seymour result, it was an open problem to know if these are defined by a finite list of forbidden minors, or recognisable in polynomial time.

Combining the theorems above, one gets the important fact that any closed-minor property is polynomially testable.

1.4 Wagner’s Conjecture

Definition 1.14

A class of graphs $C$ is said to be minor closed if for every graph $G \in C$ and every minor $H$ of $G$, $H \in C$.  

8
Definition 1.15

If $C$ is a minor closed class of graphs, a graph $G$ is a **bound** for $C$ if $G$ is not in $C$ but every strict minor of $G$ is.

Proposition 1.16

Let $C$ be a minor closed class, and $X$ be its (possibly infinite) set of bounds. Then, $G \in C \iff G$ does not contain any graph of $X$ as a minor.

Therefore, testing if a certain graph $G$ belongs to $C$ is exactly testing if $G$ contains one of the minor-minimal graphs with respect to $C$.

Here is a table describing the set of minor-minimal graphs for certain classes.

<table>
<thead>
<tr>
<th>Graph Class</th>
<th>Minor minimal graphs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forests</td>
<td>triangle</td>
</tr>
<tr>
<td>Union of Paths</td>
<td>triangle, claw</td>
</tr>
<tr>
<td>Planar</td>
<td>$K_5, K_{3,3}$</td>
</tr>
<tr>
<td>Toric</td>
<td>$\geq 16629$ (but finite)</td>
</tr>
</tbody>
</table>

We are now ready to state the celebrated conjecture formulated by Wagner in 1937:

**Conjecture 1.17 (Wagner)**

Every minor closed class of graphs has a **finite** set of bounds.

Let us mention here that high density implies the existence of a large minor; there exists theorems with better bounds, but we are only interested here in the fact that such a bounds exists.

**Theorem 1.18**

Every graph with average degree at least $2^{r-2}$ contains $K_r$ as a minor.

**Proof.** By induction on $r$. Let $G$ be a graph of average degree at least $2^{r-2}$. Therefore $|E(G)|/|V(G)| \geq 2^{r-3}$. Let $H$ be minimal amongst all minors of $G$ such that $|E(H)|/|V(H)| \geq 2^{r-3}$. It implies that when one contracts an edge in $H$, one must loose at least $2^{r-3}$ edges (otherwise the inequality would still be satisfied, and $H$ would not be minor minimal). Hence, for any $xy$ edge of $H$, $x$ and $y$ have at least $2^{r-3}$ common neighbours. In otherwords, if $x$ is a vertex in $H$, then the minimum degree in its neighbourhood is at least $2^{r-3}$, so by induction it contains a $K_{r-1}$ minor, which yields with $x$ the desired $K_r$ minor. \qed

### 1.5 Exercises

**Exercise 1.1.** Prove that a graph $G$ is a forest if and only if it does not contain $C_3$ as a minor.

**Exercise 1.2.** Show that the $(3 \times 3)$-grid has a $K_4$-minor.

**Exercise 1.3.** Let $S$ be a set of $k$ vertices in a $k$-connected graph $G$, where $k \geq 2$. Prove that there is a cycle in $G$ which includes all the vertices of $S$.

**Exercise 1.4.** Prove that $G$ is bipartite if and only if $G$ does not contain any odd circuit as a subgraph. Prove that deciding if a given graph is 2-colourable is in $P$. 

9
Exercise 1.5.

1. Give an example of $G$ and $H$ such that $G$ contains $H$ as a minor but not as a topological minor.

2. Prove that if $H$ has max degree 3 then $G$ contains $H$ as a minor if and only if it contains it as a topological minor.

Exercise 1.6. Prove that every graph with average degree at least $2^{r-2}$ contains the complete graph $K_r$ as a minor.

Let us mention here that high density implies the existence of a large minor; there exists theorems with better bounds, but we are only interested here in the fact that such a bounds exists.

**Proof.** By induction on $r$. Let $G$ be a graph of average degree at least $2^{r-2}$. Therefore $|E(G)|/|V(G)| \geq 2^{r-3}$. Let $H$ be minimal amongst all minors of $G$ such that $|E(H)|/|V(H)| \geq 2^{r-3}$. It implies that when one contracts an edge in $H$, one must lose more than $2^{r-3}$ edges (otherwise the inequality would still be satisfied, and $H$ would not be minor minimal). Hence, for any $xy$ edge of $H$, $x$ and $y$ have at least $2^{r-3}$ common neighbours. In otherwords, if $x$ is a vertex in $H$, then the minimum degree in its neighbourhood is at least $2^{r-3}$, so by induction it contains a $K_{r-1}$ minor, which yields with $x$ the desired $K_r$ minor. □

Exercise 1.7. The goal of this exercise is to prove a nice theorem of Thomassen (1981): if $G$ is a 3 connected graph on at least 5 vertices, then there exists an edge $e$ in $G$ such that $G/e$ is still 3-connected. This Theorem is often useful when one wants to prove results by induction (you first reduce the question to 3-connected graphs, then use this theorem: this is typically the case for Kuratovski’s theorem 1.5).

- Assume that $e=xy$ is an edge of $G$ such that $G/e$ is not 3-connected. Prove that there exists a vertex $z$ such that $\{x, y, z\}$ is a separator of $G$.

- Prove Thomassen’s result.

Exercise 1.8. Prove that any minor closed class of graphs is characterised by a list (not finite :) ) of forbidden minors.

Exercise 1.9. Prove that the following problems have an algorithm in time $O(f(k))n^3$.

- **$k$-leaf Spanning Tree**
  
  **Input:** A graph $G$.
  
  **Output:** TRUE if there exists in $G$ a spanning tree $T$ with at least $k$ leaves.

- **$k$-Feedback vertex set**
  
  **Input:** A graph $G$.
  
  **Output:** TRUE if there exists $k$ vertices in $G$ that intersect every cycle of $G$.

- **$k$-Vertex Cover**
  
  **Input:** A graph $G$.
  
  **Output:** TRUE if there exists a set $S$ of at most $k$ vertices in $G$ such that every edge of $G$ is incident to at least one vertex of $S$.

10
1.6 Connectivity

Two vertices $u$ and $v$ in a graph $G$ are said to be connected if $u$ and $v$ are the endpoints of some path which is a subgraph of $G$. Being connected is easily seen to be an equivalence relation. Its classes are called connected components. A graph is connected if it has only one connected component. In a connected graph $G$, a set of vertices $S$ is a separator if $G \setminus S$ is disconnected. Furthermore, $S$ separates $x$ from $y$ if $x$ and $y$ belong to two different connected components of $G \setminus S$. A graph without circuits as subgraphs is called a forest. If in addition it is connected, it is called a tree.

A graph is said to be $k$-connected if no subset $S$ of size $|S| < k$ is a separator of $G$. The following duality result due to Menger is one of the cornerstones of graph theory.

**Theorem 1.19 (Menger,1927,[18])**

> Let $G$ be a graph and $x \neq y$ be vertices of $G$. Then the minimum number of vertices whose deletion separates $x$ from $y$ is equal to the maximum number of disjoint paths between $x$ and $y$.

**Proof.** See [7].

**Corollary 1.20**

$G$ is $k$-connected if and only if between any pair of two vertices there exists $k$ disjoint paths.

Menger’s theorem being a consequence of max flow min cut theorem, it is polynomial to decide whether there exists $k$ disjoint paths between two given sets of $k$ terminals. But it is very important that the problem becomes NP-complete if the set of endpoints of the paths are specified, i.e., if $(s_1, s_2, \ldots, s_k)$ and $(t_1, t_2, \ldots, t_k)$ are given vertices, decide if there exists $k$ disjoint paths $P_i$ from $s_i$ to $t_i$. 

Chapter 2

Digression on Coloration and Planarity

This chapter was not entirely treated during the courses, I include it for curious readers.

2.1 Planar graph and minors

Kuratowski’s Theorem 1.5 says that a graph is planar if and only if it does not contain $K_{3,3}$ and $K_5$ as topological minors. A way to prove this is to prove first that in this particular case, one can replace ”topological minor” by ”minor”, because the proof is easier to make with the notion of ordinary minors.

**Proposition 2.1**

A graph contains $K_5$ or $K_{3,3}$ as a minor if and only if it contains $K_5$ or $K_{3,3}$ as a topological minor.

To prove this, one first notice that $K_{3,3}$ is not a problem thanks to Theorem 1.9. So the only thing left to prove to prove Proposition 2.1 is to prove that a planar graph that contain $K_5$ as a minor contains either $K_5$ or $K_{3,3}$ as a topological minor (left as Exercise).

Because of the previous discussion, proving Kuratowski’s Theorem relies on proving the following.

**Theorem 2.2**

A graph $G$ is planar if and only if it does not contain $K_5$ or $K_{3,3}$ as a minor.

**Proof.** (Sketch) Since the minor of a planar graph is minor (why?), and since $K_5$ and $K_{3,3}$ are not planar (why?), it is enough to prove the ”if” part of the theorem. Assume there exists a graph $G$ which is not planar but does not contain any of these two graphs as a minor. Take this $G$ minor minimal for this property, we can first prove that we can assume $G$ to be 3-connected. Then we use a result of Thomassen (Exercise 1.7) to contract an edge $e$ while preserving 3-connectivity. By minimality of $G$, $G/e$ must be planar. By considering the embedding and the vertex coming from the contracted edge, one can find the desired minor.
2.2 Chromatic Number and 4 Colour Theorem

The origin of planar graphs comes from geographic maps. Given such a map, one defines a graph with vertex set being the set of countries, and with an edge between two countries sharing a border. The fundamental question about planar graphs was asked by the south-african mathematician and botanist Francis Guthrie, who was then a student in London.

**Conjecture 2.3 (Guthrie, 1852)**

Using a set of only four colours, it is always possible to colour the countries such that two countries sharing a border have different colours.

The first attempts resulted in ‘proofs’ with mistakes (notably, Kempe 1879 and Tait 1880), and this conjecture only became a theorem in 1976, due to Appel and Hakken ([2],[3]). Their proof is particular because it is not human-checkable. Indeed there is a list of 1,936 reducible configurations that need to be checked by a computer.

Let us define more formally the chromatic number of a graph and give a list of basic properties.

**Definition 2.4**

Let $G = (V, E)$ be a graph.

- A $k$-coloration is a function $c : V \mapsto \{1, \ldots, k\}$ such that for every edge $xy \in E$, $c(x) \neq c(y)$.
- The minimum $k$ such that there exists a $k$-coloration of $G$ is called the **chromatic number** of $G$, denoted $\chi(G)$.

Note that graphs with chromatic number 1 are exactly stable sets, graphs with chromatic number 2 are called **bipartite graphs** (see Exercice ??).

Guthrie’s conjecture states that the dual of a planar graph has chromatic number at most, where the **dual** of an embedded planar graph is the graph whose vertices are faces of the original graph (i.e. connected components of $\mathbb{R}^2 \setminus G$) and edges correspond to faces sharing a border (one can argue, that depending on the choice of the embedding, the definition of the dual is not unique, which is true, note that thanks to a Theorem of Whitney, every 3-connected planar graph has only one planar embedding on the sphere, and hence its dual is defined uniquely) In fact every planar graph is a dual of some planar graph (namely, its dual), so Guthrie’s conjecture is really asking for the following.

**Theorem 2.5 (Appel Hakken, 1975 [2])**

Every planar graph has chromatic number at most 4.

In terms of complexity, computing the chromatic number is an NP-complete problem, even in restricted cases as the following theorem proves (see [13] for a proof).

**Theorem 2.6**

Decide if $G$ is 3-colourable is NP complete, even for the class of planar graphs with maximum degree 4.
Let us now describe some upper bounds. By a greedy algorithm it is easy to prove that the chromatic number satisfies

\[ \chi \leq \Delta + 1 \]

where \( \Delta \) denotes the maximum degree. One just chooses any linear ordering of the vertices of the graph, and assuming vertices \( v_1, v_2, \ldots, v_i \) are coloured, choose for vertex \( v_i \) the smallest colour not assigned to its already coloured neighbours. It yields clearly a \( \Delta+1 \) colouring. (Note that 2 different orderings can give radically different number of colours used. In fact, for every graph, there exists an ordering such that the greedy algorithm applied to this ordering yields an optimal colouring. For these two questions, see Exercice 2.1.) This greedy bound is tight for odd cycles or cliques. In fact Brooks proved that these are the only connected graphs for which this equality holds.

**Theorem 2.7 (Brooks)**

If \( G \) is a connected graph and is neither a clique nor an odd cycle, then \( \chi \leq \Delta \).

**Proof.** See [7] or [10]. \( \square \)

A way to generalise the idea of the greedy algorithm is the notion of degeneracy, which is also linked to vertex colourings. A graph \( G \) is \( k \)-degenerate if there exists an enumeration \( v_1, \ldots, v_n \) of the vertices of \( G \) such that:

\[ \forall i, d_{\{v_1, \ldots, v_{i-1}\}}(v_i) \leq k. \]

**Proposition 2.8**

\( G \) \( k \)-degenerate \( \Rightarrow \) \( \chi(G) \leq k + 1 \)

Here are two trivial inequalities linking chromatic number with independence number and clique number.

**Proposition 2.9**

For every graph \( G \) on \( n \) vertices, \( \chi(G) \geq n/\alpha(G) \) and \( \chi(G) \geq \omega(G) \).

A class of graphs is said to be \( \chi \)-bounded if there exists a function \( f \) such that \( \chi(G) \leq f(\omega(G)) \) for every graph \( G \) in the class. The most famous example is the class of Berge graphs which are the graphs that do not contain as a induced subgraph the odd holes (induced cycles of odd length at least 5) nor their complement. A conjecture of Berge from the 60’s was to prove that these graphs are exactly the graphs for which every subgraph satisfies \( \chi = \omega \). This was one of the most studied questions in graph theory for a long time until its resolution in 2002 by Chudnovsky, Robertson, Seymour and Thomas. This sole topic could be the subject of several courses, we refer the interested reader to [7] for more informations and bibliography.

The class of all graphs is not \( \chi \)-bounded as there exists graphs with arbitrarily big chromatic number, which do not contain any triangle as a subgraph. Several explicit constructions are known to prove this fact (Zykov, Mycielski, Blanche Descartes), see Exercise 2.4 for a description of Mycielski’s construction. In fact a celebrated result of Erdős of 1959, asserts that something quite stronger is true. The **girth** (in French **maille**) of a graph is the length of a shortest circuit of \( G \). Being triangle free implies that the girth is at least 4. In fact, if the girth of \( G \) is greater than \( g \), then it means that the balls of radius \( \lfloor g/2 \rfloor \) around any vertex are isomorphic to trees (which are of chromatic number 2). Hence the following result is quite unexpected at first, and so was this proof, which was one of first proofs to show the power of the so called Probabilistic Method, see [12].
**Theorem 2.10**

For any integer $k$, there exists a graph $G$ with chromatic number at least $k$ and girth at least $k$.

**Proof.** (Sketch) Take a random graph $G_{n,p}$ ($n$ vertices, every edge is taken independently with probability $p$) with $p = n^{-(k-1)/k}$, and set $t = 2 \log(n)/p$. Then it can be shown that almost surely $\alpha(G) \leq t$ and almost surely the number of cycles of length less than $k$ in $G$ is no more than $n/2$.

Therefore, by deleting at most $n/2$ vertices from $G$ (one from each small cycle), one gets a graph $G'$ with girth at least $k$, and with $\alpha(G') \leq t$. Then $\chi(G') \geq |V(G')|/\alpha(G) \geq n/2t = n^{1/k}/8 \log n$. By choosing $n$ large enough, one gets that $\chi(G') \geq k$. □

### 2.3 5-colorability of Planar Graphs

Here we are going to show a simple proof for 5 colorability of planar graphs. First we need the following very famous theorem.

**Theorem 2.11 (Euler, 1752)**

If $(G, \phi)$ is a plane graph with $n$ vertices, $m$ edges, $f$ faces, and $c$ connected components, then

$$f - m + n - c = 1$$

**Proof.** Induction on the number of edges. □

Note that Euler’s formula implies that the number of faces depends only on the number of vertices and edges, and not on the embedding of the graph.

Euler’s formula implies the following crucial inequality on the number of edges in a planar graph. It uses a classical combinatorial proof method, called double counting.

**Proposition 2.12**

Every simple planar graph on $n$ vertices has at most $3n - 6$ edges.

**Proof.** Double counting of the number of edges. Use the fact that each face has length at least 3. □

Note that this gives an alternate proof of the non planarity of $K_5$ and $K_{3,3}$. But more importantly it implies that planar graph are 5-degenerate, and hence 6-colourable. We give here the proof of the 5-colorability, which is based on the approach Kempe used in his false proof of 1879 of the 4-colour Theorem.

**Theorem 2.13 (Heawood, 1890)**

Every planar graph is 5 colourable

**Proof.** The proof works by induction on the number of vertices. We can assume the graph is triangulated. By Proposition 2.12, there is a vertex of degree 5 and by induction we can assume that it has degree exactly 5 and that its 5 five neighbours use the 5 colours. Let $C = v_1v_2v_3v_4v_5$ be the facial cycle of $G \setminus u$ whose vertices are the neighbours of $v$, where $v_i$ receives color $i$. Consider the subgraph of $G$ induced by colours 1 and 3. If $v_1$ and $v_3$ belong to different connected components of this graph, then we can switch...
colours 1 and 3 inside the components containing \( v_1 \). We still get a good colouring of \( G \setminus H \) and we can give the color 1 to \( u \). Therefore we can assume there is a path from \( v_1 \) to \( v_3 \) using colours 1 and 3, and similarly with \( v_2 \) and \( v_4 \) with colours 2 and 4. But these path should intersect, which is not possible because of the colours, contradiction. \( \Box \)

### 2.4 Hadwiger’s Conjecture

In the previous section we saw that large density implies the existence of a big clique minor. In 1942 Hadwiger made a conjecture of this kind but with a condition on the chromatic number. This question is still open and is probably one of the most important open problems in graph theory today.

**Conjecture 2.14** (*Hadwiger, 1943*)

*If a graph \( G \) does not contain \( K_k \) as a minor, it is \( k - 1 \) colourable.*

This conjecture is trivial for \( k = 1, 2, 3 \), not difficult for \( k = 4 \) (see Exercise 2.14). Wagner proved (1964) that the case \( k = 5 \) is equivalent to the 4 colour theorem. Robertson Seymour and Thomas proved the case \( k = 6 \) in 1993 (very difficult, also proved using the 4-colour theorem). The conjecture remains widely open and some mathematicians believe it might be false. It appears to be hard to prove also for big values of \( k \). For example for \( k = n/2 \), the conjecture implies (thanks to Proposition 2.9) that if \( \alpha(G) = 2 \), then \( G \) should contain a \( K_{\lfloor n/2 \rfloor} \) minor. Proving \( \lceil n/3 \rceil \) is not difficult, but no one has succeeded in bridging this gap (see Exercise 2.13). To finish, note that the equivalent conjecture with ”topological minor” instead of ”minor”, formerly known as Hajós conjecture, was proven to be false, first by counterexample of Catlin (1979) and then by a result of Erdős and Fajtlowicz (1981) who totally demolished the conjecture by proving that almost every graph is a counterexample. Their proof is another example of the use of the probabilistic method (see for example [7]).

### 2.5 Exercises

**Exercise 2.1.** Consider the graph \( G_n \) obtained from a complete bipartite \( K_{n,n} \) by a removing a perfect matching. Prove that depending on the linear ordering chosen to apply the greedy algorithm, one can use 2 colours on \( n \) colours.

Prove that for any graph, there exists a linear ordering of its vertices such that the greedy algorithm gives an optimal colouring.

**Exercise 2.2.** Prove Proposition 2.9

**Exercise 2.3.** Degeneracy

- Prove Proposition 2.8

- Prove that \( G \) is \( k \)-degenerate if and only if every subgraph of \( G \) contains a vertex of degree at most \( k \).

- Characterise the 1-degenerate graphs.

- For \( k \) fixed, is is polynomial to decide if \( G \) is \( k \)-degenerate?

**Exercise 2.4.** Consider the following recursive construction of graphs \( M_n \), for \( n \in \mathbb{N} \).
• $M_1 = K_2$.

• Assume $M_n$ is given, let us construct $M_{n+1}$. Start with a copy of $M_n$. For every vertex $x \in V(M_n)$, add a new vertex $x'$ such that $N(x') = N(x)$. Now add a last new vertex $u$ adjacent to all the new vertices $x'$.

Prove that $\chi(M_n) = n - 1$ and that every graph in the family is triangle free.

**Exercise 2.5.** Show that the Petersen graph is not planar by proving it contains a subdivision of $K_{3,3}$.

**Exercise 2.6.** Check Euler’s formula for stable sets, trees and forests.

**Exercise 2.7.** Prove that there exists only 5 platonic solids, i.e. polyhedra with every face of the same length, and every vertex contained in the same number of faces.

**Exercise 2.8.** Prove that there exists only 5 platonic solids, i.e. polyhedra with every face of the same length, and every vertex contained in the same number of faces.

**Exercise 2.9.** Prove that every planar graph is the union of 3 forests.

**Exercise 2.10.** Define the crossing number $cr(G)$ of a graph $G$ as the least number of crossing of edges in a plane embedding of $G$.

• Prove that $cr(K_5) = cr(K_{3,3}) = 1$ and that $cr(K_6) = 3$.

• Prove that If $G$ is a graph with $n \geq 3$ vertices and $m$ edges, then $cr(G) \geq m - 3n + 6$

Let us mention a stronger result on this topic. It was proven first independently by Ajtai et al. (1982, see [1]) and Leighton (1983, see [16]). The probabilistic proof shown here is due to Alon (for a very good monograph on the probabilistic method, see Alon and Spencer’s book [12]).

**Theorem 2.15 (Ajtai et al., Leighton)**

If $G$ is a graph with $n$ vertices and $m \geq 4n$ edges, then

$$cr(G) \geq \frac{1}{64} \frac{m^3}{n^2}$$

**Proof.** Consider a planar embedding $\phi$ of $G$ with $cr(G)$ crossings. Let $S$ be a random subset of $V$ obtained by choosing each vertex of $G$ independently with probability $p = 4n/m$. Let $H$ be the graph induced by the vertices in $S$. Define the random variables $X,Y,Z$:

• $X$ is the number of vertices of $H$
• $Y$ the number of edges of $H$
• $Z$ the number of crossings of $H$ in the embedding given by $\phi$.

from Corollary ??, we have $Z \geq Y - 3X + 6$, and by linearity of expectation $E(Z) \geq E(Y) - 3E(X) + 6$. Now, $E(X) = pn$, $E(Y) = p^2m$, and $E(Z) = p^4cr(G)$. Hence,

$$cr(G) \geq \frac{pm - 3n}{p^3} = \frac{n}{(4n/m)^3} = \frac{1}{64} \frac{m^3}{n^2}$$

$\square$
Exercise 2.11. Prove Proposition 2.1.

Exercise 2.12. A graph $H$ is an algebraic dual of a graph $G$ if there is a bijection $\phi : E(G) \rightarrow E(H)$ such that a subset $C$ of $E(G)$ is cycle of $G$ if and only if $\phi(C)$ is a bond of $H$.

- Prove that every planar graph has an algebraic dual and that $K_5$ and $K_{3,3}$ don’t.
- A theorem due to Whitney (1932) states that a graph is planar if and only if it has an algebraic dual. Prove this theorem using Kuratowski’s Theorem.

Exercise 2.13. Prove that if $\alpha(G) = 2$, then $G$ admits a $K_{\lfloor n/3 \rfloor}$ minor.

Exercise 2.14. Prove that if every vertex has degree at least 3, then the graph contains a subdivision of $K_4$. (Hint: you can prove something stronger: if every vertex but maybe one has degree $\geq 3$, then the conclusion still holds).
Chapter 3

Minor Closed Classes and Well Quasi Orders

3.1 Wagner’s Conjecture - Well Quasi Orders

Recall that a class of graphs is minor-closed if every minor of a graph of the class is also in the class. In Chapter 1, we discussed Robertson and Seymour proof of a conjecture of Wagner:

Conjecture 3.1 (Wagner)

For every minor closed class $C$, there exists a finite set of graphs $F_C$ (often called obstructions) such that a graph belongs to $C$ if and only if it does not contain (as a minor) any graph in $F_C$.

Let us discuss here what can these obstructions be. For a given minor closed class $C$, a graph $H$ is said to be a bound if $G$ is not in $C$ but every strict minor of $G$ is. Note that if $H$ is a bound, since it is not in $C$ it must contain any obstruction so it must be itself an obstruction. The following easy proposition tells us that the set of bounds is in fact a sufficient set of obstructions.

Proposition 3.2

Let $C$ be a minor closed class, and $B$ be its (possibly infinite) set of bounds. Then $G \in C$ if and only if $G$ does not contain any graph of $X$ as a minor.

Proof. If $H$ not in the class then either it is minimal or it contains $H'$ not in the class. We can repeat the argument, and since there exists no infinitely decreasing sequence of graphs, every graph not in the class admits one of the bound as a minor. □

(This uses the fact that there exists no infinite decreasing sequence of graphs for the minor order - such partial orders are called well founded.)

As said before, it implies that testing if a certain graph $G$ belongs to $C$ is exactly testing if $G$ contains one of the minor-minimal graphs with respect to $C$. Here is a table describing the set of minor-minimal graphs for certain classes.

<table>
<thead>
<tr>
<th>Graph Class</th>
<th>Minor minimal graphs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forests</td>
<td>triangle</td>
</tr>
<tr>
<td>Union of Paths</td>
<td>triangle, claw</td>
</tr>
<tr>
<td>Planar</td>
<td>$K_5, K_{3,3}$</td>
</tr>
<tr>
<td>Toric</td>
<td>$\geq 16629$ (but finite)</td>
</tr>
</tbody>
</table>
So another way of stating Wagner conjecture would be to say: **Every minor closed class of graphs has a finite set of bounds.** Note that by definition one bound cannot be the minor of another. Using the terminology of partially ordered sets, they form an **antichain**: a set of pairwise not comparable elements. So a way to prove Wagner conjecture would be to prove that there exists no infinite antichain for the minor relation on graphs. In fact this equivalent as we will show now.

**Definition 3.3**

A partial order \( \preceq \) defined on a set \( X \) is a **well quasi order** (WQO) if there is no infinite decreasing sequence and no any infinite antichain.

A infinite sequence that is either decreasing or an antichain will always be called a **bad sequence**. A wqo is hence defined as a partial order with no bad sequences. Note that in the case of graphs, there cannot be an infinite decreasing sequence, so the only possible bad sequence would be an infinite antichain.

**Proposition 3.4**

Wagner’s conjecture 3.1 is equivalent to say that the class of all graphs with the minor relation is a wqo.

**Proof.** Assume that the minor relation is a wqo. consider a class \( C \) that is minor closed. Let \( F_C \) be the class of graphs minimally (for the minor relation) not in \( C \). Then \( F_C \) is an antichain, so it is finite, and it is easy to see that \( G \) is in \( C \) if and only if \( G \) does not contain any graph in \( F_C \) as a minor. Now assume that Wagner’s conjecture is true. Assume there exists a bad sequence of graphs \( G_n \). Let \( C \) be the class that do not contain any of the \( G_n \) as a minor. It is minor closed, hence there exists a finite list \( (H_i) \) such that \( G \) is in \( C \) iff it does not contain any of the \( H_i \) as a minor. So every \( G_i \) must contain one of these graphs as a minor. By pigeonhole principle, there exists \( G_i \) and \( G_j \) that contain the same \( H_k \). But conversely, \( H_k \) is not in \( C \) so by definition it must contain one of the \( G_n \) as a minor. by transitivity, this contradicts the fact that the \( G_n \) form an antichain.

\( \Box \)

In the next section we will try to understand some of the ideas behind the proof of Wagner’s conjecture by proving similar but (much) easier results, the main of which being a theorem due to Kruskal saying that trees are well quasi ordered for the minor relation.

### 3.2 Words, Paths and trees

Before stating the first result of this section about wqo, we prove the following proposition, that will be of use all through the section.

**Proposition 3.5**

Let \( (X, \leq) \) be a partially ordered set and \( (x_i)_{i \in \mathbb{N}} \) be any sequence. Then this sequence has an infinite subsequence that is either increasing, or decreasing or an antichain.

**Proof.** Let \( (x_i) \) be any sequence. Start with \( x_1 \), and consider

- \( A_1 = \{ j, \ j > 1 \text{ and } x_1 \leq x_j \} \)
- \( B_1 = \{ j, \ j > 1 \text{ and } x_1 \leq x_i \} \)
- \( C_1 = \{ j, \ j > 1 \text{ and } x_1 \text{ and } x_j \text{ are incomparable} \} \)
If \( A_1 \) is infinite we say that \( x_1 \) is of type \( A \) and delete all elements that are not in \( A_1 \). If not, but \( B_1 \) is infinite, say that \( x_1 \) is of type \( B \) and delete all elements that are not in \( B_1 \). Finally in the last case, say that \( x_1 \) is of type \( C \) and delete all vertices not in \( C_1 \).

Up to extracting a subsequence and renaming, we can assume no elements were deleted, so that all \( x_i \) with \( i \geq 2 \) were in \( A_1 \), or \( B_1 \), or \( C_1 \). We do this sequentially of \( x_2 \), then \( x_3 \), ... . At each step, we define \( A_i, B_i, C_i \) as

- \( A_i = \{ j, \ j > i \text{ and } x_j \leq x_i \} \)
- \( B_i = \{ j, \ j > i \text{ and } x_j \leq x_i \} \)
- \( C_i = \{ j, \ j > i \text{ and } x_i \text{ and } x_j \text{ are incomparable} \}

and at each step we define the type of \( x_1 \) to be one of \( A, B, C \) depending on which is infinite. Then we extract by keeping only the elements in the infinite set.

Eventually we have a type for each element of the sequence (which is in fact an subsequence of the original sequence). Now there must be a type with infinitely number of elements and to each type clearly corresponds one of the three possible type of infinite subsequence. \( \square \)

From this we deduce the following which gives equivalent conditions for being a wqo.

**Corollary 3.6**

Let \((X, \leq)\) be a partially ordered set. The three assertions are equivalent

1. \((X, \leq)\) is a wqo
2. from every sequence \((x_i)_{i \in \mathbb{N}}\) one can extract an infinite increasing subsequence.
3. from every sequence \((x_i)_{i \in \mathbb{N}}\) one can extract \( i < j \) such that \( x_i \leq x_j \).

This will be useful: in order to prove that a given partial order is a wqo, we will only prove the third statement, but when we use the fact that an order is a wqo (for example in a proof by induction), we can use the second statement which is (in appearance) much stronger.

Let us illustrate this by proving that the set of words on a finite alphabet is wqo for the subword relation.

**Proposition 3.7**

Assume \( X \) is a finite set, and define \( X^* \) as the set of finite sequence of elements of \( X \). Define a partial order \( \preceq \) on \( X^* \) by \( u \preceq v \) if \( u \) is a subword of \( v \) (\( u \) can be obtained from \( v \) by deletion of letters). Then \((X^*, \preceq)\) is a wqo.

**Proof.** We do a proof by induction on the size of \( X \). Assume the opposite, and let \( w_1, w_2, \ldots \) be a bad sequence. It implies in particular that \( w_1 \not\equiv w_i \) for every \( i \geq 2 \). This will help us to gain some structure on the \( w_i \). Let \( w_1 = x_1 x_2 \cdots x_i \) and define the function \( f : X^{(i)} \to \{0, 1, \ldots, k\} \) by \( f(w) = \max\{i, \ x_1 x_2 \cdots x_i \leq w\} \) (by convention \( f(w) = 0 \) if no such \( i \) exists). There must be an integer \( 0 \leq j \leq k - 1 \) such that \( \{i, \ f(w_i) = j\} \) is infinite. To simplify notations and avoid considering a subsequence, we assume in fact that \( f(w_i) = j \) for all \( i \).

Look now at a word \( w \) with \( f(w) = j \); It means it can be decomposed in \( w = u_1 x_1 u_2 x_2 \ldots u_j x_j u_{j+1} \), for all \( i \), and \( u_i \) are words such that \( x_i \not\equiv u_i \) for all \( i \) between 1 and \( j + 1 \). Each \( u_i \) thus belong to the set of words on the alphabet \( X_i = \{x \in X, \ x_i \not\equiv x\} \). By
induction these sets of words are wqo for $\leq$, then by extracting an infinite increasing subsequence in the $u'_i$, then extracting from this subsequence another infinite increasing subsequence for $u'_i$, etc., one can get an infinite increasing subsequence in the $w_i$, which contradicts the hypothesis.

One can remark that words with the subword relation correspond to paths (as graphs) with labels on the vertices. A word is a subword of another, if the first path admits a subdivision that is a subpath of the second, with the same labels. In other words, we have proven that labelled paths are wqo for the topological minor with labels order. We will soon prove this for trees.

Higman Theorem replaces the word “finite” by WQO:

**Proposition 3.8 (Higman, 1952, [14])**

Let $(X, \leq)$ be a wqo. Extend this partial order to $X^*$ by $x_1x_2 \ldots x_k \leq y_1y_2 \ldots y_l$ if there exists an increasing injection $f : \{1, \ldots, k\} \to \{1, \ldots, l\}$ such that for every $i$, $x_i \leq y_{f(i)}$. Then $(X^*, \leq)$ is also a wqo.

Note that it implies the previous one: if $X$ is finite, it suffices to consider on $X$ the partial order where no two elements are comparable - it clearly defines a wqo.

**Proof.** Assume by contradiction the existence of bad sequences, and choose a minimal bad sequence (MBS) in the following sense: if $w_1, w_2, \ldots, w_i$ are defined, $w_{i+1}$ is chosen as minimal (with respect to its length), amongst the words $w$ for which there exists a bad sequence starting with $w_1, w_2, \ldots, w_i, w$. (such a word always exists as there is no infinite decreasing sequence here).

Let $w_i = x_iw'_i$. Assume the $(w'_i)$ contains a bad subsequence $w'_{\phi(i)}$, where $\phi$ is an increasing function. Consider the sequence $w_1, w_2, \ldots, w_{\phi(1)-1}, w'_{\phi(1)}, w'_{\phi(2)}, w'_{\phi(3)}, \ldots$. It is easy to see that it is also a bad sequence, contradicting the minimality of $(w_i)$. Therefore $(w'_i)$ contains no bad subsequence. It implies that it contains an increasing subsequence, but now since $X$ is a wqo, it exists two indices $i$ and $j$ from this subsequence such that $x_i \leq x_j$, which yields $w_i \leq w_j$, contradicting the hypothesis.

In fact we will not use exactly the previous proposition, but the following that deals with finite subsets. The proof is almost identical.

**Proposition 3.9 (Higman, 1952, [14])**

Let $(X, \leq)$ be a wqo. Extend this partial order to $X^{(<\omega)}$ (finite subsets of $X$) by $A \leq B$ if there exists an injection $f : A \to B$ such that for every $a \in A$, $a \leq f(a)$. Then $(X^{(<\omega)}, \leq)$ is also a wqo.

We are now ready to move on to the next step, trees. A famous result from Kruskal is the following.

**Theorem 3.10**

Finite trees are well quasi ordered for the topological minor relation

Notice that it is stronger than the same statement with the standard minor relation. Here we are going to prove an even stronger result, because we are going to consider topological minor on labelled rooted trees: we consider finite trees whose nodes are labelled by elements from a wqo $(X, \leq)$. As in the case of words for Higman’s Lemma, we extend this partial order on $W$ to a partial order on those labelled rooted trees. Let $T$ and $T'$ two such trees.
We say that $T \preceq T'$ if there exists a subdivision $T''$ of $T$ with an isomorphism $\phi$ between $T''$ to a subgraph of $T'$ such that

- $\phi$ preserves the tree order: if $x$ is an ancestor of $y$ in $T$, then $\phi(x)$ is an ancestor of $\phi(y)$ in $T'$.
- $\phi$ preserves the labels: for every vertex $x$ of $T$, $\text{label}(x) \preceq \text{label}(\phi(x))$ (we don’t consider for this the vertices of $T'' \setminus T$ obtained by subdivisions, they don’t have any label anyway).

If $X$ has cardinality one, this is exactly the topological minor relation for rooted trees. Note also that this partial order restricted to paths is exactly the partial order defined on words in Higman’s Theorem 3.8. The proof given below is due to Nash-Williams.

**Theorem 3.11 (Kruskal)**

Consider finite labelled trees and the partial order $\preceq$ as in the above paragraph. Then it defines a well quasi order.

**Proof.** Assume there exists a bad sequence of trees for this order, and consider a sequence $T_n$ of trees that is a minimal bad sequence, which is defined the following way: if $T_0, T_1, \ldots, T_i$ are defined, $T_{i+1}$ is chosen as a minimal tree (with respect to its size) such that there exists an infinite bad sequence starting with $T_0, \ldots, T_{i+1}$.

Denote by $r_i$ the root of $T_i$, and $\mathcal{U}_i$ the family of trees obtained by removing $r_i$ from $T_i$. Let $\mathcal{U}$ be the union of all $\mathcal{U}_i$.

We first prove that $\mathcal{U}$ is well quasi ordered. Assume for contradiction that $Z$ contains a bad sequence. Up to taking a subsequence, we can assume that there exists a bad sequence $U_{\phi(i)}$ such that for all $i$, $U_{\phi(i)} \in \mathcal{U}_{\phi(i)}$. Consider the sequence $T_0, T_1, \ldots, T_{\phi(0)-1}, U_{\phi(0)}, U_{\phi(1)}, U_{\phi(2)}, \ldots$. It is easy to see that it is a bad sequence, which contradicts the minimality of the $T_i$.

Therefore, this set $\mathcal{U}$ is well quasi ordered, and by Theorem 3.9, the set of finite subsets of $\mathcal{U}$ is also wqo. Therefore the sequence $(\mathcal{U}_i)_{i \in \mathbb{N}}$ admits an infinitic increasing subsequence $(\mathcal{U}_{\phi(i)})_{i \in \mathbb{N}}$ (remember that the order on subsets is the one defined in Theorem 3.9). Now as in Higman’s Theorem, we look at the sequence of labels of the roots $r_{\phi(i)}$. Since the set of labels is wqo, there exists $i < j$ such that $l(\alpha_{\phi(i)}) \preceq l(\alpha_{\phi(j)})$. Together with $\mathcal{U}_{\phi(i)} \preceq \mathcal{U}_{\phi(j)}$, this gives $T_i \preceq T_j$, and our contradiction.

3.3 Exercises

**Exercise 3.1.** For each of the following classes, decide if it is minor closed or not. If not, try to describe the smallest minor closed class containing it: cliques, paths, cycles, graphs of max degree $k$?

**Exercise 3.2.** For each of these, say if it is a wqo.

- $(\mathbb{N}, \leq)$,
- $(\mathbb{R}, \leq)$,
- $(\mathbb{N}^2, \leq)$ where $(x, y) \preceq (x', y')$ if $x \leq x'$ and $y \leq y'$,
(\mathcal{G}, \preceq) \text{ where } \mathcal{G} \text{ is the class of all graphs, and } G \preceq H \text{ if } G \text{ (induced) subgraph of } H

\bullet \text{ Finite trees ordered by subgraph relation.}

\bullet (\mathcal{G}, \prec) \text{ where } G \prec H \text{ if } G \text{ topological minor of } H

**Exercise 3.3.** In this exercise, the partial order and containment relation considered is the induced subgraph relation. \( P_k \) denotes the induced path on \( k \) vertices.

1. What is the class of \( P_2 \)-free graphs? Is it wqo?
2. What is the class of \( P_3 \)-free graphs? Is it WQO?
3. Prove that is \( G \) is \( P_4 \)-graphs then either \( G \) is disconnected or its complement is disconnected.

Deduce from this, that the class of \( P_4 \)-free graphs is WQO.

4. Prove that this is not true for \( P_5 \)-free graphs.

**Exercise 3.4.** In this exercise, the partial order and containment relation considered is the subgraph relation.

\bullet \text{ Prove that in any connected graphs, paths of maximum length are pairwise intersecting.}

\bullet \text{ Prove that if there exists an infinite antichain for the subgraph relation, then there exists an infinite antichain containing only connected graphs.}

\bullet \text{ Prove that the class of connected graphs that do not have } P_l \text{ as a subgraph is wqo for the subgraph order by proving the stronger result : the class of } P_k \text{-free graphs with vertices labelled with a finite number of labels is wqo for labelled subgraph relation } (H \preceq G \text{ if there is an label preserving isomorphism from } H \text{ to a subgraph of } G).
Chapter 4

Tree Width

4.1 Definitions

Definition 4.1

- Let \( G \) be a graph. A tree decomposition of \( G \) is a pair \((T, W)\) where \( T \) is a tree and \( W = (W_t)_{t \in V(T)} \) a collection of subsets of \( V(G) \) satisfying:
  - For every \( u \in V(G) \), \( T_u = \{ t \in V(T) \mid u \in W_t \} \) induces a connected subgraph of \( T \).
  - For every edge \( uv \in E(G) \), \( T_u \cap T_v \neq \emptyset \).
- The width of a tree decomposition is \( \max_{t \in V(T)} |W_t| - 1 \).
- The tree width of a graph \( G \), denoted \( \text{tw}(G) \), is the minimum width of a tree decomposition of \( G \).

Equivalently, a tree decomposition of \( G \) is a tree \( T \) along with a collection of subtrees \( T_v \), one for each vertex of \( G \), with the condition that \( T_u \) and \( T_v \) intersect if \( uv \) is an edge of \( G \). Note that is not a equivalence, it is possible that \( T_u \cap T_v \neq \emptyset \) even if \( uv \notin E(G) \) (this will be the case for chordal graphs).

Here is a key lemma regarding subtree intersection; by analogy with Helly’s Theorem on convex subsets of \( \mathbb{R}^d \), this property is often called Helly property of subtrees of a tree.

Lemma 4.2

Let \( F \) be a collection of pairwise intersecting subtrees of a given tree \( T \). Then \( \cap_{T \in F} T \neq \emptyset \).

Proof. If not, for each vertex \( t \) of the tree, there is a subtree in \( F \) that does not intersect this vertex, and therefore is contained in one of the components of \( T \setminus t \). One edge incident to \( t \) corresponds to this component, orient this edge out from \( t \). One gets this way an orientation of some edges of \( T \) such that each vertex has exactly one outgoing edge. Since there are less edges than vertices in a tree, there must be an edge oriented both ways, which results in two non intersecting subtrees in \( F \). Contradiction.

Corollary 4.3

Let \( G \) be a graph and \( K \) be complete subgraph of \( G \). In any tree decomposition \((T, W)\) of \( G \), there exists a vertex \( t \) of \( T \) such that \( K \subset W_t \). In particular, \( \text{tw}(G) \geq \omega(G) - 1 \).
Or course tree decompositions, even optimal ones, are not unique. Let us try to define a way to somehow minimise an optimal decomposition. Assume there are two adjacent vertices \( s \) and \( t \) of \( T \) such that \( W_s \subset W_t \). Then we can contract the edge \( st \) of \( T \) to a new vertex \( r \), and define \( W_r = W_s \). It is trivial to check that this defines a valid tree decomposition with width no larger than the original one. By repeating this operation we obtain the following proposition.

**Proposition 4.4**

For every graph \( G \), there exists a tree decomposition of width \( \text{tw}(G) \) such that for every edge \( st \in E(T) \), \( W_s \not\subset W_t \) and \( W_t \not\subset W_s \). In particular, for every leaf \( f \in V(T) \), there exists a vertex \( u \in V(G) \) such that \( T_u = \{ f \} \).

**Theorem 4.5**

In every graph \( G \), there exists a vertex of degree at most \( \text{tw}(G) \).

**Corollary 4.6**

\[ \chi(G) \leq \text{tw}(G) + 1 \]

### 4.2 Chordal graphs

In this section, we show how tree decompositions are related to chordal graphs.

**Definition 4.7**

A graph \( G \) is **chordal** if it has no induced subgraph isomorphic to a cycle of length at least 4.

We will prove that chordal graphs are exactly the graphs for which tree decomposition satisfy the property that \( uv \in E(G) \) if and only if \( T_u \) and \( T_v \) have non empty intersection (we say that \( G \) has a subtree intersection model).

**Lemma 4.8**

If \( G \) is a chordal graph and \( S \) is a minimal separator of \( G \), then \( S \) is a clique.

Using this lemma and the Helly property for chordal graphs, one can prove the following theorem, that gives two characterisation of chordal graphs.

**Theorem 4.9**

The following statement are equivalent

1. \( G \) is chordal
2. Every induced subgraph of \( G \) contains a **simplicial vertex**, i.e. a vertex whose neighbourhood is a clique.
3. \( G \) is the intersection graph of a family of subtrees of a tree.

Using this characterisation, one gets this equivalent definition of treewidth.

**Proposition 4.10**

\[ \text{tw}(G) = \min\{ \omega(G') - 1, \text{ G subgraph of } G' \text{ and } G' \text{ is chordal} \} \]
4.3 Minors

Proposition 4.11

If $H$ is a minor of $G$, then $\text{tw}(H) \leq \text{tw}(G)$

**Proof.** It is enough to prove that the three operations of vertex deletion, edge deletion and edge contraction cannot increase treewidth. So starting from an optimal tree decomposition of $G$, we find one for the new graph without increasing the size of the bags.

- for $G \setminus e$, do nothing
- for $G \setminus v$, just remove $T_v$.
- for $G/e$, where $e = uv$ : the new vertex is called $w$. Delete $T_u$ and $T_v$, set $T_w = T_u \cup T_v$

□

Corollary 4.12

The class of graphs of treewidth at most $k$ is closed under taking minors.

Therefore, using the theorem of Robertson and Seymour, we know that it is defined by a finite number of excluded minors. In fact, this result is not a consequence of their theorem, but one of its steps, as this result can be proven directly using the ideas of the proof of Kruskal Theorem 3.11.

Theorem 4.13

The class of graphs of treewidth at most $k$ is well quasi order for the minor relation

**Proof.** Admitted. Uses a variant of Kruskal Theorem. □

Corollary 4.14

The class of graphs of treewidth at most $k$ has a finite number of bounds.

**Proof.** of Corollary 4.14 If $G$ is a bound for the class of $\text{tw} \leq k$, then $\text{tw}(G) = k + 1$ (why?). So the set of bounds is an antichain contained in the class of graphs of treewidth at most $k + 1$. Hence it is finite by Theorem 4.13. □

Let us try to describe the bounds for small values of $k$.

Theorem 4.15

- $\text{tw}(G) \leq 1 \iff G$ is a forest $\iff G$ does not contain $K_3$ as a minor
- $\text{tw}(G) \leq 2 \iff G$ does not contain $K_4$ as a minor

**Proof.** The first item is trivial, and for the second, one direction is easy : since $\text{tw}(K_4) = 3$, graphs of tree width at most 2 are indeed $K_4$-minor free.

So let us prove that if $G$ is $K_4$ minor free, then it has treewidth at most 2. First we can assume that $G$ is not 3 connected. Indeed, assume by contradiction that $G$ is 3-connected. Let $C$ be a minimum cycle in $G$. If $G = C$, then $\text{tw}(G) = 2$. If not, there
exists \( x \in G \setminus C \), and by Menger’s Theorem, there exists three disjoint paths from \( x \) to \( C \). Using these paths and the cycle, one gets a \( K_4 \) minor.

Therefore \( G \) has a separator of size at most 2. Assume \( G \) has a separator \( \{a, b\} \) of size 2 where \( ab \notin E \). It is easy to see that one can add the edge \( ab \) without creating a \( K_4 \) minor. Hence we can add the edge \( e = ab \) and the graph \( G + e \) still has np \( K_4 \)-minor, and \( G + e \) has a clique separator of size 2. So in all cases we can assume that \( G \) has a separator \( S \) which is a clique (of size 1 or 2), and we can glue he decompositions of the components of \( G \setminus e \) to conclude by induction that \( G \) has treewidth at most 2. \( \square \)

This proofs shows the role of separators with treewidth. The operation of gluing graphs along cliques is called clique sum and will be described in the next section.

One could hope for a general result \( tw(G) = k \) iff \( G \) does not have a \( K_{k+2} \) minor. Unfortunately, this fails for \( k = 3 \). There exists graph with no \( K_5 \) minor (and as we will see soon, planar graphs) with arbitrarily high treewidth.

Due to the following theorem, there are in fact four bounds for tree width at most 3. \( O \) is the octahedron, \( W_8 \) is the cycle on 8 vertices where all edges linking diametrically opposite nodes are added, and \( C_5 \times K_2 \) is the pentagonal prism (two \( C_5 \) joined by a perfect matching).

**Theorem 4.16**

\[
tw(G) \leq 3 \iff G \text{ does not contain one of the four following graphs as a minor : } K_5, W_8, O \text{ and } C_5 \times K_2.
\]

**4.4 Digression : Hadwiger Conjecture**

We have already proven these two sets of inequalities

\[
\begin{align*}
\omega(G) & \leq \chi(G) \leq tw(G) + 1 \\
\omega(G) & \leq \omega_m(G) \leq tw(G) + 1
\end{align*}
\]

where \( \omega_m(G) \) denotes the largest integer \( k \) such that \( G \) has a \( K_k \) minor. Now, note that Hadwiger Conjecture 2.14, can be formulated exactly as \( \chi(G) \leq \omega_m(G) \). Let us examine rapidly the easy cases.

It is trivial that \( \omega_m(G) \leq 2 \iff G \text{ is a forest } \Rightarrow \chi(G) \leq 2. \)

As we have seen before \( \omega_m(G) \leq 3 \iff tw(G) \leq 2 \Rightarrow \chi(G) \leq 3 \) by the above inequalities.

Note that the next case, \( \omega_m(G) \leq 4 \Rightarrow \chi(G) \leq 4 \) implies the Four Colour Theorem since planar graphs are \( K_5 \)-minor free. In fact it is equivalent (and hence true), thanks to a structural characterisation of graphs with no \( K_5 \) minor due to Wagner.

**4.5 Separators**

The purpose here is to prove that indeed tree decomposition ‘decompose’ the graph. Let us state first an easy but fundamental result.

**Proposition 4.17**

\[
\begin{align*}
\text{Let } (T, W) \text{ be a tree decomposition of } G \text{ and } t_1 t_2 \text{ be an edge of } T \text{ and denote by } S \text{ the set of vertices } W_{t_1} \cap W_{t_2}. \text{ For } i = 1, 2, \text{ define } T_i \text{ as the connected component of } T \setminus t_1 t_2 \text{ containing } t_i, \text{ and } G_i \text{ the subgraph of } G \text{ induced by } \bigcup_{t \in T_i} (W_t \setminus S). \text{ Then there are no edges between } G_1
\end{align*}
\]
and $G_2$. This can be stated also the following way

**Theorem 4.18**

If $H$ is a connected subgraph of $G$, then for any tree decomposition $(T, W)$ of $G$, $\bigcup_{v \in V(H)} T_v$ induces a subtree of $T$.

Conversely, we have the following result about cutsets that induce complete graphs.

**Proposition 4.19**

Let $G$ be a graph with a clique cutset $S$ and let $(X_i)_{i \in I}$ be the connected components of $G \setminus S$. Define $G_i$ to be the graph induced by $X_i \cup S$. Then $\text{tw}(G) = \max_{i \in I}(\text{tw}(G_i))$.

**Definition 4.20**

Let $G_1$ and $G_2$ be two graphs and $K_1$ a clique of $G_1$, $K_2$ a clique of $G_2$ with $|K_1| = |K_2|$. If $G$ is a graph obtained by identifying vertices of $K_1$ and $K_2$, and then removing some edges of this clique, then $G$ is a **clique sum** of $G_1$ and $G_2$.

Again Proposition 4.19 can be restated in terms of clique sums.

**Proposition 4.21**

If $G$ is a clique sum of $G_1$ and $G_2$, then $\text{tw}(G) \leq \max(\text{tw}(G_1), \text{tw}(G_2))$.

We have seen that $K_4$-minor free graphs are exactly graphs with treewidth at most 2. Hence they have separators of size at most 1 and 2. In fact they are exactly the graphs obtained from $K_2$ by operations called series and parallel (see figure below) and are often called for these reasons **series-parallel** graphs.

Now we are going to give a proof that planar free graphs are not bounded for treewidth, by proving that grids can have arbitrarily high treewidth. To do so we begin with the following proposition, which gives a lower bound on the treewidth in terms of minimum good separator.

**Proposition 4.22**

Let $G$ be a graph of tree width $k$. Then there exists a subset $X \subset V(G)$ such that

- $X$ is a cutset of $G$
- $X$ has size at most $k + 1$
- no connected component of $G \setminus X$ has size larger than $|V(G)|/2$

**Proof.** First we can assume the graph is connected, for we can always apply this result on the biggest connected component. Now let $(T, W)$ be an optimal tree decomposition and $t_1t_2$ be an edge of $T$. Let us use the notations of Proposition 4.17, and orient $t_1t_2$ towards $t_i$ if $G_i$ contains more than $n/2$ vertices. The edge cannot be oriented in both directions otherwise the would be no vertices in $W_i \cap W_{t_2}$ and $G$ would not be connected. Therefore there exists a vertex $t \in T$ such that no edge incident to $t$ is oriented out from $t$. The bag $W_i$ is the desired set of vertices. \[\square\]
Now we are able to prove that the $2k \times 2k$-grid has treewidth at least $k$ (in fact we will prove in section 4.6, that the treewidth is $2k$).

**Corollary 4.23**

The treewidth of the grid $G_{2k,2k}$ is at least $k$.

**Proof.** Assume the contrary, use Proposition 4.22 to get a good cut of size at most $k$. There are at least $k$ rows and 1 column that this set misses. Since these rows and columns form a connected set of vertices, they are all in the same component which has therefore size at least $2k^2 \geq |G|/2$. Contradiction. □

**Corollary 4.24**

The class of planar graph has unbounded treewidth.

### 4.6 Duality - Cops and Robbers

As we have seen at the beginning of this chapter, if $G$ is chordal, then its treewidth is precisely equal to $\omega(G) - 1$. Therefore, to certify that some decomposition is optimal, it is sufficient in that case to show in the graph some clique of the appropriate size. For a general graph $G$, one can use the largest order of a clique minor, or the largest $k$ such that a $k,k$ grid is a minor, as a lower bound on $\text{tw}(G)$. However, these bounds can be far from the exact value. For example grids have arbitrarily large treewidth, but no clique minor of order more than 4. The purpose of this section is to provide a dual notion for treewidth, that is something that certifies an exact lower bound.

Note that in the proof of Proposition 4.22, the crucial property is that the set of connected subgraphs of size at least $n/2$ satisfy the following: either they intersect, or there is an edge from one to the other (the whole graph is connected). For any collection of such connected subsets that satisfy this, there exists one bag of the decomposition that intersects all of them. This is precisely the notion behind the dual of treewidth.

**Definition 4.25**

- We say that two connected subgraphs of $G$ **touch** if they have non empty intersection or if they are joined by an edge.
- A **bramble** of $G$ is a collection $\mathcal{B}$ of connected subgraphs that are pairwise touching.
- A **transversal** of a bramble $\mathcal{B}$ is a set of vertices of $G$ that has non empty intersection with each element of $\mathcal{B}$.
- The **order** of a bramble $\mathcal{B}$ is the minimum size of a transversal of $\mathcal{B}$.
- The **bramble number** of $G$, denoted $\text{bn}(G)$, is the maximum order of a bramble of $G$.

This notion was introduced by Seymour and Thomas and is the dual notion of treewidth, as the following theorem proves.

**Theorem 4.26 (Seymour and Thomas, 1993)**

For every graph $G$, $\text{bn}(G) = \text{tw}(G) + 1$
This theorem is a sort of minmax theorem (in fact maxmin=minmax).

As we have explained before the definitions, to prove the lower bound one just needs to mimic the proof of Proposition 4.22, which is concerned by the bramble formed by connected subgraphs of size at least \(|V(G)|/2\). In a few paragraphs, we will give a different proof of this using a 2 player game one the graph. The upper bound is harder to prove. In the next chapter we will prove an approximate theorem, namely that \(tw(G) \leq 4bn(G)\).

Let us illustrate the lower bound with the grid \(G_{n,n}\). We already know that its tree width is at most \(n\). So with the previous theorem, we only need to exhibit some bramble of order \(n + 1\). Denote by \(x_{ij}\) the vertex on \(i\)-th row and \(j\)-th column. Define a bramble whose sets are \(\{A, B\} \cup \{C_{ij}, 1 \leq i, j < n\}\), where:

- \(A = \{x_{i,1}, 1 \leq i \leq n\}\), the last row,
- \(B = \{x_{1,j}, 1 \leq j < n\}\) the last column minus its last element,
- \(C_{ij} = \{x_{k,j}, 1 \leq k < n\} \cup \{x_{ik}, 1 \leq k < n,\}\).

It is easy to check that this constitutes a bramble of order \(n + 1\).

Now we want to do an alternate proof of the lower bound by introducing an third invariant \(cn(G)\), and prove the two inequalities \(tw(G) + 1 \geq cn(G)\) and \(cn(G) \geq bn(G)\). This will come from the fact that \(cn(G)\) will be defined through a two player game, where tree decomposition corresponds exactly to strategies for one player, and brambles correspond exactly to strategies for the second player.

Let us define what a cops and robber game on a graph is. In that kind of two player game, one is controlling several cops, the other one is controlling the robber, and the goal of the first player is to capture the robber. There are many variants of this game (see [ ] for a state of the art), depending on where the cops/robber can move, on how they move, at which speed, or on the type of game (turn by turn or simultaneous). Here we are describing a variant that was introduced by Seymour and Thomas in [23].

In this variant, cops are standing on vertices of the graph, and at each turn a fraction of them can move by helicopter and land on any vertex of the graph. The robber sees the helicopter approaching and can instantly move at infinite speed to any other vertex along a path of a graph. The only constraint is that he is not permitted to run through a vertex occupied by some cop. The cops win if they capture the robber, so they win if at some point they occupy all vertices adjacent to the position of the robber, and an extra cop lands by helicopter on the robber.

For a graph \(G\), the **cop number** of \(G\), denoted \(cn(G)\) is the smallest number of cops to ensure the capture of the robber.

**Proposition 4.27**

\[
\text{cn}(G) \leq tw(G) + 1
\]

**Proof.** Let \((T, W)\) be a tree decomposition of \(G\) with width \(tw(G)\). Put every cop one the vertices of some bag \(W_t\). The robber, if it escapes has to be in some vertex appearing only in the bags of some component of \(T \setminus t\). Let \(t'\) the neighbour of \(t\) in \(T\) in the direction of this component. Thanks to Proposition 4.17 \(W_t \cap W_{t'}\) separates the component containing he robber form the rest of the graph. At the next move, the player
controlling moves cops in $W_t \setminus W_r$ to occupy all of $W_r$. He now keeps on applying this strategy until it reaches some leaf of the tree and the robber cannot escape.

**Proposition 4.28**

\[ \text{bn}(G) \leq \text{cn}(G) \]

**Proof.** Let $\mathcal{B}$ be a bramble of order $\text{bn}(G)$ and assume there are only $\text{bn}(G) - 1$ cops. Let $C$ be the set of initial positions of the cops. By definition there exists a set $X \in \mathcal{B}$ such that $X \cap C = \emptyset$. The robber moves to some vertex $x \in X$. After that, the game really begins, a fraction of the cops moves and lands, so that the new set occupied by the cops is $C'$. Again there exists $X' \in \mathcal{B}$ such that $X' \cap C' = \emptyset$. During their flight the only occupied vertices are $C \cap C'$ so $X \cup X'$ is entirely free of cops, and the robber can freely move from $X$ to $Y$ and this strategy can be applied for ever. □

### 4.7 Treewidth and Planar Graphs

To begin this section, let us try to begin a proof of Wagner’s Conjecture. Suppose $(G_n)_{n \in \mathbb{N}}$ is a sequence of graphs. We want to prove that there exists $G_i$ and $G_j$ with $i < j$ and $G_i$ is a minor of $G_j$. If there exists $G_i$ such that $G_i$ is a minor of $G_i$, then we have won. If not, we know that $G_2, G_3, \ldots$ all belong to the class of $G_1$-minor free graphs. If we can prove that this class has bounded treewidth, then we have won by Theorem 4.13. We are going to see that this is true if $G_1$ is planar. Theorem 4.31 below, states that the class of $H$-minor free graphs has bounded treewidth if $H$ is planar (and in fact if and only if).

First, note that every planar graph is the minor of some grid

**Proposition 4.29**

*If $G$ is a planar graph, then there exists $k$ such that $G$ is a minor of $G_{k,k}$*

Therefore, for a given $H$ there exists a $k(H) \in \mathbb{N}$ such that the class of $H$ minor free graphs is included into the class of $G_{k,k}$-free graphs. It remains to prove that these have bounded treewidth. The following theorem, often called the Grid Minor Theorem, achieves this.

**Theorem 4.30**

*For any $k$, the class of $G_{k,k}$-minor free graphs has bounded treewidth.*

Equivalently if a graph has large enough treewidth, it will contain any grid, as a minor. The proof of this result will not be given here, because it is long and difficult, we refer to the reader to [10]. In 2013, Chekuri and Chuzhoy, [8], proved the first polynomial bound, that is there exists a $C > 0$ such that the class of $G_{k,k}$-minor free graphs has treewidth at most $k^C$.

We are now able to prove the aforementioned result.

**Theorem 4.31**

*The class of $H$-minor free graphs has bounded treewidth if and only if $H$ is planar*

**Proof.** Suppose first that $H$ is not planar. Since the class of planar graph is minor-closed, this implies that the class of $H$-minor free graphs contains all planar graphs, and has therefore unbounded treewidth, thanks to Corollary 4.24.
Conversely, if $H$ is planar, then by Proposition 4.29 the class of $H$ minor free graphs is contained in the class of $G_{k,k}$-minor free graphs, for $k$ sufficiently large. But now Theorem 4.30 tells us that this class has bounded treewidth. \hfill \Box

With this result and the discussion at the beginning of this section, we now have.

**Corollary 4.32**

The class of planar graphs is wqo for the minor relation.

### 4.8 Wagner’s Conjecture

To get the proof of Wagner’s conjecture, we would like to imitate the proof of Theorem 4.32. So we start by considering a sequence $(G_n)_{n \in \mathbb{N}}$ that is supposedly a counterexample. As before we can assume that no graph $G_i$ with $i \geq 2$ has $G_1$ as a minor. And as before we would like to use this fact to get some structure for these graphs. In fact it is sufficient to get a structure theorem for all graphs not containing $K_l$, for $l$ fixed as a minor. We already saw two theorems characterising this for $l \leq 5$.

It would be nice to have this for every $l$, unfortunately, we can guess that it would be very complicated; So the idea is to prove approximate characterisations.

Let us start with a technical definition. If $C$ is a cycle, a **vortex** on $C$ is defined the following way:

- select a collection of arcs $A_1, A_2, \ldots, A_l$ on $C$ so that each vertex is in at most $k$ arcs.
- For each arc we add a vertex $v_j$ that is linked to some vertices of $A_i$.
- we can also add edges $v_i v_j$ if $A_i \cap A_j \neq \emptyset$.

Now let us define a class $G_k$

- Start with a surface of genus at most $k$ with a graph $G$ embedded in it so that each face is homeomorphic to a disc.
- Add at most $k$ vortices on faces of $G$
- Add at most $k$ vertices (apexes) linked arbitrarily to the rest of the graph.

Such are graphs are called “nearly embeddable” in a surface of genus $k$. Eventually define $L_k$ as the closure (for clique sums) of the class $G_k$. Now the fundamental structure theorem can be given.

**Theorem 4.33**

For every graph $H$, there exists an integer $k$ such that the class of $H$-minor free graphs is included in $L_k$.

Eventually this says that $H$-minor free graphs have on the outside a tree like structure (due to the sequence of clique sums operations), and inside these are graphs of bounded genus plus a bounded number of perturbations (the vortices and apex vertices). Very (very) roughly, graphs of bounded genus are taken care of by induction on the genus, and then Kruskal’s Theorem’s proof is adapted to deal with the tree structure of clique sums operations.
4.9 Rooted disjoint path problem and minor detection

Several times in this course we have seen the following connectivity problem:

**Input:** A graph $G$, an integer $k$ and two subsets of vertices $X$ and $Y$ be two subsets of vertices of size $k$

**Output:** TRUE if there exists $k$ vertex disjoint paths from $A$ to $B$?

As we have seen, this problem of connectivity has got a dual by Menger’s theorem in terms of separators (and hence is in co-NP) but is in fact polynomially solvable by max-flow techniques.

Nevertheless, the smilingly similar problem is NP-complete for $k \geq 2$:

**Input:** A graph $G$, an integer $k$, and two subsets of vertices $X = \{x_1, x_2, \ldots, x_k\}$ and $Y = \{y_1, y_2, \ldots, y_k\}$

**Output:** TRUE if there exists disjoint paths $P_1, P_2, \ldots, P_k$, such that $P_i$ is a path from $x_i$ to $y_i$.

In the Graph Minor series of papers, Robertson and Seymour gave a polynomial algorithm for fixed $k$, and as was discussed in Chapter 1, this gives a polynomial algorithm to decide if a fixed $H$ is a minor of some input graph $G$.

Let us describe rapidly the ideas behind the algorithm for rooted disjoint path problem mentioned above. One is the following: in some situations it is possible to prove that there exists vertex $v$ which is irrelevant to the existence of these paths, and if we can find it, we delete it from the graph, and apply the algorithm inductively.

A simple case is the case where $G$ contains a big clique. Suppose it contains a clique of size $2k$. Then apply Menger to $X \cup Y$ and $C$, if we can find the $2k$ disjoint paths we can answer YES by rerouting inside the clique. If not, there is a separator $S$ of size $s < 2k$ and in fact there exist a set of paths $\mathcal{P}$ from $X \cup Y$ to $C$ of size $s$. By considering minimal such paths we know that there exists a subset $C'$ of $C$ which don’t appear on these paths. Now if there exists $k$-disjoint routed paths from $X$ to $Y$, then by rerouting them using the paths in $\mathcal{P}$ we know that there also exists a solution which avoids $C'$. These vertices are irrelevant to our problem. So we can delete those vertices and try again, and thus we can assume that the graph does not contains a big clique.

What Robertson and Seymour prove is that it works similarly if $G$ contains a very large grid minor. If the desired paths exists, there is surely a way to reroute the parts of the paths that go through the grid so that some vertex in the ‘middle’ of the grid is not used, and this will be our irrelevant vertex.

Thanks to Theorem 4.30 we know that if the treewidth is large enough, such a big grid minor exists. If not, there we are in the case of bounded treewidth, and we will see in the next chapter that dynamic programming approaches permits us to solve the problem. So the difficult part is to effectively and efficiently find the irrelevant vertex in the case of big treewidth. This splits into two parts: first if the graph contains a large clique minor, this is not too difficult, then if no large clique minor exists, Robertson and Seymour use their structure Theorem 4.33 and the existence of a large grid minor to find the irrelevant vertex.

As mentioned in the first chapter, this theorem implies Theorem 1.11 saying that minor detection is polynomial, and hence it implies that any closed under minors property can be decided in polynomial time, since such a property is defined by a finite list of excluded minors. For example this theorem solves the question for linklessly embeddable graphs (graphs that can be embedded in three-dimensional spaces, so that two cycles can always be pulled apart, that is they don’t entwine like links in a chain), a problem that was open before the Graph Minors papers.
4.10 Tree Decompositions of Planar Graphs

In this section we sum up several properties of tree decomposition of planar graphs, that will be useful for designing algorithms on such graphs.

We first start by studying a particular sub class of planar graphs. A planar graph is said to be outerplanar, if there exists an planar embedding of the graph such that all vertices appear on the outer face.

It is easy to see that this is a minor closed class, and since $K_4$ is not outerplanar, any outerplanar graph has treewidth at most 2. Another forbidden minor is complete bipartite $K_{2,3}$ and it is possible in fact to prove the following characterisation.

**Theorem 4.34**

$G$ is outerplanar if and only if $G$ does not contain neither $K_4$ nor $K_{2,3}$ as a minor.

To prove the difficult implication : if $G$ contains neither $K_4$ nor the complete bipartite $K_{2,3}$ as a minor, then $G$ is outerplanar, one can use directly Kuratowski’s Theorem. One can also prove it directly, see Exercise 4.17.

Now for every planar graph, one can define how close it is to an outerplanar graph that we explain now. An embedding of a graph is $k$-outerplanar if removing the vertices on the outer face results in a $k-1$-outerplanar embedding (a 0-outerplanar embedding is the empty embedding). A graph is $k$-outerplanar if it admits a $k$-outerplanar embedding.

Thus every planar graph is $k$-outerplanar for a certain value of $k$. We will first prove that this implies that the treewidth of $G$ is at most $3k$.

**Theorem 4.35**

Let $G$ be a planar graph of radius at most $R$, i.e. there exists a vertex $r$ and a spanning tree $T$ rooted at $r$ with depth at most $R$. Then $\text{tw}(G) \leq 3R - 1$ and it is possible to get a tree decomposition of width at most $3R - 1$ in polynomial time.

For a proof see Exercise 4.18

Now consider a $k$-outerplanar graph. So it can be thus decomposed into $k$ successive layers $L_1, \ldots, L_k$ such that $L_1$ is the outerface, $L_2$ is the outerface of $G \setminus L_1$, etc... Add one vertex on the outerface connected to every vertex of this face, and add also edges between layers so that every vertex of $L_i$ is adjacent to at least one vertex of $L_{i-1}$. This can be done while preserving planarity, and the resulting graph $G'$ has treewidth at least that of $G$, and one can apply the previous Theorem to get that $G$ has treewidth at most $3k - 1$.

Using this fact, one can now prove the following result. For a proof, see Exercise 4.20

**Theorem 4.36**

There exists $C > 0$ such that if $G$ is a planar graph on $n$ vertices, then $\text{tw}(G) \leq C \sqrt{n}$.

As a corollary, one can get a famous theorem of Lipton and Tarjan.

**Theorem 4.37**

Every planar graph on $n$ vertices admits a separator $S$ of size $O(\sqrt{n})$ such that no component of $G \setminus S$ has more than $n/2$ vertices.

In fact, these theorems are equivalent in the sense that using Lipton and Tarjan Theorem, one can build a tree decomposition of width $O(\sqrt{n})$.  

35
4.11 Digression: Erdős-Posa Property

One natural question about cycles in a graph $G$ is to determine the minimum number of vertices that one needs to delete in order to get an acyclic graph. Equivalently, this is the minimum size of a set $X$ of vertices such that $X \cap C \neq \emptyset$ for every circuit $C$ of the graph. Such a set is often called a transversal of the family of circuits, and we denote here the minimum size of a transversal by $\tau_C(G)$.

Clearly if there exists in our graph $k$ pairwise vertex-disjoint circuits, then a transversal contains at least $k$ vertices. Therefore, if $\nu_C(G)$ denotes that maximum number of pairwise vertex-disjoint circuits, we have just proven that $\nu_C(G) \leq \tau_C(G)$.

A classical result by Erdős and Posa proves that for $\nu_C$ fixed, $\tau_C$ cannot be arbitrarily large.

**Theorem 4.38 (Erdős-Posa, ’65)**

There exists a function $f : \mathbb{N} \rightarrow \mathbb{N}$, such that for any graph $G$:

$$\tau_C(G) \leq f(\nu_C(G)).$$

Now, note the following: being a transversal of the family of circuits is the same as being a transversal of the family of subgraphs of $G$ having $K_3$ as a minor (we call these subgraphs extensions of $K_3$ in $G$). Therefore one can try to generalise Erdős and Posa’s theorem and try to ask the same question with any graph $H$ instead of $K_3$. A graph $H$ will then be said to have the **Erdős-Posa property** if there exists as in Theorem 4.38 a function $f$ for the family of extensions of $H$. Using the results of the previous sections, one can settle this question, these $H$ are exactly the planar graphs.

**Theorem 4.39**

A graph $H$ has the Erdős-Posa Property if and only if $H$ is planar.

We prove it in the case where $H$ is a connected graph. For a complete proof one can check [].

**Proof.** Assume first that $H$ is connected planar and let $G$ be a graph such that the maximum number of disjoint extensions of $H$ in $G$ is equal to $k - 1$. Then if we denote by $H_k$ the graph which is the disjoint union of $k$ copies of $H$, this exactly means that $G$ contains $H_{k-1}$ as a minor but not $H_k$, $H_k$ being planar, by Theorem 4.31 there exists an integer $t_k$ such that, if a graph has treewidth at least $t_k$, then it contains $H_k$ as a minor. Let us define recursively the function $f$ by $f(0) = 0$ and $f(k) = 2f(k - 1) + t_k$ for any $k \geq 1$. We are going to prove by induction on $k$ that for any graph $G$, either $G$ contains $k$ disjoint extensions of $H$ in $G$ or by removing at most $f(k)$ vertices from $G$ we get a $H$-minor free graph.

Assume the result is true up to $k - 1$. By what precedes, if $G$ has treewidth at least $t_k$, then it has $k$ disjoint extensions of $H$, so we can assume $tw(G) < t_k$. Fix an optimal tree decomposition $(T, W)$ of $G$. For an edge $t_1t_2$ of $T$, let $T_1$ and $T_2$ be the connected components of $T \setminus t_1t_2$ and denote for $i = 1, 2$

$$G_i := \cup_{t \in T_i} (W_i \setminus W_t)$$

We now orient every edge $t_1t_2$ towards $t_i$ if $G_i$ contains $H$ as a minor. Note that doing so every edge can receive one, two or no orientation.

If every edge gets at most one direction, then it means some vertex $t$ of the tree has no outgoing edge, which means that $W_t$ intersects every extension of $H$ in $G$, and since
\[|W_i| \leq t_k \leq f(k),\] we get our desired transversal. If this is not the case, some edge gets two orientations, which means both \(G_1\) and \(G_2\) contain \(H\) as a minor. Since they are disjoint, if one of these two graphs contains \(k - 1\) disjoint extensions of \(H\), then \(G\) contains \(k\) disjoint copies and we are done. Therefore we can assume that \(G_1\) and \(G_2\) do not contain \(H_{k-1}\) as a minor. By induction there exists \(f(k - 1)\) vertices in each graph whose removal leaves each graph without any \(H\) minor. By taking these two sets plus \(W\), one gets the desired transversal of size at most \(f(k)\).

Now let us prove the converse and assume \(H\) is non planar. Let \(\Sigma\) be a surface in which \(H\) embeds, such that \(\Sigma\) is minimal genus wise. This minimality ensures that any two drawings of \(H\) intersect. Now one can construct a graph by embedding \(2k + 1\) vertex disjoint copies of \(H\) in \(\Sigma\) such that no point of \(\Sigma\) is contained in more than 2 of these copies. Transform every crossing of edges into a vertex of degree 4. The resultant graph \(G\) has no transversal of size \(k\), but every two copies of \(H\) intersect.

\(\Box\)

4.12 Exercises

Exercise 4.1. Prove Corollary 4.3

Exercise 4.2. Determine the treewidth of a path, a tree, a complete bipartite graph, the cube.

Exercise 4.3.

1. Prove that if \(G\) contains (as a subgraph) a complete bipartite with parts \(A\) and \(B\), then in every tree decomposition there exists a bag that contains \(A\) or a bag that contains \(B\).

2. Prove that if \(x\) and \(y\) are two vertices that are joined by \(k + 1\) vertex disjoint paths, then in every tree decomposition of \(G\) of width at most \(k\), there exists a bag containing both \(x\) and \(y\).

3. Prove that \(G\) is \(K_{2,3}\) minor free then \(tw(G) \leq 3\).

Exercise 4.4. Prove that tree decompositions given by Proposition 4.4 satisfy \(V(T) \leq V(G)\). Prove also Theorem 4.5 and its Corollary.

Exercise 4.5. Show that graphs \(G\) of treewidth at most \(k\) with \(k \geq 1\) have strictly less than \(k|V(G)|\) edges.

Exercise 4.6. Prove that If \(H\) is a subdivision of \(G\), then \(tw(H) = tw(G)\)

Exercise 4.7. Prove Proposition 4.19. Prove also that the same is true for the chromatic number, and for the function which associates to a graph the size of its largest clique minor.

Exercise 4.8. Prove that, if \(O\) denotes the octahedron graph, \(tw(O) = 4\). Prove that for any edge \(e\) of \(O\), \(tw(O \setminus e) = 3\).

Exercise 4.9. Show that for each \(W \subset V\), and any tree decomposition \(T\) it holds that either \(W\) is a subset of some node of \(T\) or \(W\) is separated by \(X_i \cap X_j\) for some edge \(X_iX_j \in E(T)\)

Exercise 4.10. Find a bramble of order \(n + 1\) for the grid \(G_{n,n}\).

Exercise 4.11. What is the cop number for stable sets, cliques, cycles?

Exercise 4.12. Prove directly that \(cn(G) = 2\) if and only if \(G\) is a forest.

Exercise 4.13. Prove directly that \(tw(G) + 1 \geq bn(G)\) without the use of the cops and robber game. You can prove that for any bramble and any tree decomposition, there exists a bag of the decomposition that is a transversal of that bramble. Relate this proof to the proof of Proposition 4.22.
Exercise 4.14. Prove that there exists an integer $C$ such that for any $K_4$-minor free graph $G$, there exists a subset $X$ of $C$ vertices in $V(G)$ such that $\lambda(G \setminus X) < \lambda(G)$, where $\lambda(G)$ denotes the maximum length of a path in $G$.

Exercise 4.15. Feedback Vertex Set

A feedback vertex set of a graph $G$ (in short FVS) is a set of vertices whose removal leaves the graph acyclic. The problem for an input graph to determine the size of a minimum FVS, denoted $\text{minFVS}(G)$ is a classical NP-Problem (in the famous list of Karp 23 NP-complete problem, see [15]).

1. Prove that for any graph, $\text{tw}(G) \leq \text{minFVS}(G) + 1$.
2. Is it true that there exists a function $f$ such that for every graph $G$, $\text{minFVS}(G) \leq f(\text{tw}(G))$?
3. Prove that if $H$ is a minor of $G$, then $\text{minFVS}(H) \leq \text{minFVS}(G)$.
4. If $k$ is a fixed integer, prove that there exist a finite family of graphs $G_k$ such that $\text{FVS}(G) \leq k$ if and only if $G$ contains as a minor some graph of $G_k$.
5. Using a theorem seen in this course give an existential proof for a $O(n^3)$ algorithm to decides for input $G$ whether $\text{minFVS}(G) \leq k$. Why does it give an FPT algorithm with parameter $k = \text{minFVS}(G)$ to determine the value of $\text{minFVS}(G)$?
6. Prove that the class of graphs that have a FVS of size at most 1 is characterised by the three following forbidden minors (the third one being disconnected : the disjoint union of two $K_3$):

![Forbidden minors](image)

Exercise 4.16. The pathwidth $\text{pw}(G)$ of a graph $G$ is defined as the minimum width of a tree decomposition $(T, W)$ of $G$ where $T$ is a path.

1. Prove that the pathwidth of the complete graph $K_n$ is $n - 1$.
2. Prove that if $H$ is a minor of $G$ then $\text{pw}(G) \geq \text{pw}(H)$
3. Prove that if $G$ is connected, there exists a path $P$ in $G$ such that $\text{pw}(G \setminus P) < \text{pw}(G)$.
4. Determine the graphs of path width 1.
5. Give a construction of a sequence of trees $T_p$ such that $\text{pw}(T_p) = p$.
6. A harder Theorem is the following :

Theorem 4.40

For every forest $T$, there exists a constant $f(T)$ such that for every graph $G$, $\text{pw}(G) \geq f(T)$ implies that $T$ is a minor of $G$.

What is the analogue Theorem for treewidth instead of pathwidth?

7. Bienstock, Robertson, Seymour et Thomas proved that this Theorem is true for $f(T) = |V(T)| - 1$. Prove that this cannot be replaced by $|V(T)| - 2$
8. Prove that this Theorem is also optimal in the sense that it is not possible to replace forest
by any larger class of graphs.

Exercise 4.17. Outerplanar Graphs (en français : Graphes Planaires Extérieurs)
A planar graph is said to be outer planar, if there exists an planar embedding of the graph such
that all vertices appear on the outer face.
1. Prove that this is a minor-closed class that contains neither $K_4$ nor the complete bipartite
$K_{2,3}$. Using Kuratowski’s Theorem, prove that this is equivalent.
2. Prove that every outerplanar graph contains a vertex of degree 2. Prove that every outer
planar graph is 3-colorable.
3. Prove that every outerplanar graph has treewidth at most 2.
4. Prove without Kuratowski’s Theorem that if $G$ contains neither $K_4$ nor the complete
bipartite $K_{2,3}$ as a minor, then $G$ is outerplanar. Hint : prove that minimal separators have size at
most 2.

Exercise 4.18. Planar Graphs of bounded Radius
The goal of this exercise is to prove the following. Let $G$ be a planar graph of radius at most $R$ :
there exists a vertex $r$ and a spanning tree $T$ rooted at $r$ with depth at most $R$. Then $tw(G) \leq R$
and it is possible to get a tree decomposition of width at most $3l$ in polynomial time.
1. Prove that one can assume that $G$ is triangulated, i.e. there exists a planar embedding of
$G$ such that every face of $G$ is a triangle.
2. Let $F$ be the set of faces of $G$ in such an embedding and let $T^*$ be the graph whose
vertices are the faces of $G$, and where $fg \in E(T^*)$ if and only if faces $f$ and $g$ share an edge
e $e \in E(G) \setminus E(T)$. Prove that $T^*$ is a forest.
3. For every face $f$ of $G$ of boundary $uvw$, denote by $X_f$ the set obtained by $\{u, v, w\}$ by
adding all vertices of $G$ which are ancestors of $u, v$ or $w$ inside $T$. Let $v$ be a vertex $G$.
Prove by induction on the depth of $v$ inside $T$ that for every $v$, $\{f \in F, v \in X_f\}$ induces a connected graph
of $T$ (start by leaves of $T$ and go up the tree).

Exercise 4.19. Treewidth of $k$-outerplanar Graphs
A planar graph is said to be outerplanar (or 1-outerplanar) if there exists a planar embedding
of $G$ where all its vertices lie on the outer face (for example, $K_4$ is planar but not outerplanar).
A planar graph is $k$-outerplanar if by deleting its outer face, one gets a $k − 1$-outerplanar graph.
Such a graph can be thus decomposed into $k$ successive layers $L_1, \ldots, L_k$.
By using the previous exercise prove that every $k$-outerplanar graph has treewidth at most $3k$.

Exercise 4.20. Treewidth of planar graphs
The goal of this exercise is to prove that for planar graphs, treewidth is of order $O(\sqrt{|V(G)|})$.
1. Prove that we cannot expect something smaller than $O(\sqrt{|V(G)|})$ for the class of planar
graphs.
2. Prove that this result implies the following theorem due to Lipton and Tarjan : ”Every
planar graph on $n$ vertices admits a separator $S$ of size $O(\sqrt{n})$ such that no component of $G \setminus S$
has more than $n/2$ vertices”.
3. Let $G$ be a graph and assume there exists a family of $S_1 \cup \ldots \cup S_t$ pairwise disjoint subsets
vertices of of $G$ satisfying :
   - there is no edge between $S_i$ and $S_j$ for any $|j − i| \geq 2$. 

39
• For every connected component $C$ of $G \setminus (S_1 \cup \ldots \cup S_t)$ there exists $i$ such that the only neighbours of $C$ are in $S_i \cup S_{i+1}$.

• For every $i$, $|S_i| \leq x$.

• $\text{tw}(G \setminus (S_1 \cup \ldots \cup S_t)) \leq t$.

Prove that $\text{tw}(G) \leq t + 2x$.

4. Prove that if $G$ is planar, $\text{tw}(G) < 2 \sqrt{6|V(G)|}$, (hint: use the previous exercise).
Chapter 5

Tree Width and Algorithms

5.1 Algorithms on Trees

We are going to start with a short section about dynamic programming on trees. Let us consider
the problem of maximum weight independent set. The input is a graph \( G = (V, E) \) whose vertices
are given weights through a function \( \omega : V \rightarrow \mathbb{R} \). We define the weight of a subset of vertices
to be the sum of the weights of its elements. The problem is now to find an independent set of
maximum weight. Of course the problem for general graphs is NP-complete (it contains the
maximum independent set problem). Nevertheless for trees, there exists an easy polynomial
algorithm for this problem.

Fix a root \( r \) arbitrarily and orient all edges away from this root. Denote by \( C(v) \) the set of
children of \( v \), by \( T(v) \) the subtree rooted at a vertex \( v \) (hence \( T(r) = T \)) and by

- \( W(v) \) denote the maximum weight of an independent set of \( T(v) \),
- \( W^+(v) \) denote the maximum weight of an independent set of \( T(v) \) containing \( v \)
- \( W^-(v) \) denote the maximum weight of an independent set of \( T(v) \) not containing \( v \)

Then we can compute inductively \( W, W^+, \) and \( W^- \), since

\[
W(v) = \max(W^+(v), W^-(v)) \\
W^+(v) = w(v) + \sum_{u \in C(v)} W^-(u) \\
W^-(v) = \sum_{u \in C(v)} W(u)
\]

Eventually, we can compute \( W(r) \) which is exactly the value of the maximum weight independent
set of \( T \).

Like this problem, many NP-complete problems become polynomial on trees. On the next
section we are going to see that this stands also for bounded treewidth graphs. That is we are
going to find algorithms whose complexity is a function \( f(t, n) \) of the treewidth \( k \) and the size of
graph \( n \), and this function will be a polynomial in \( n \).

5.2 Algorithms for bounded-treewidth graphs

In this section we will assume that the graph \( G \) has treewidth at most \( t \), AND we will assume
that we are given a nice tree decomposition \( T \) of width at most \( t \) (we will explain later how to
get rid of this second assumption later, but is important that we do not take care of the time complexity needed to compute this decomposition). As we will see through examples, many NP-hard problems become polynomial when restricted to bounded treewidth graphs.

**Theorem 5.1**

*Computing maximum weight independent set can be solved in time* $O(f(t).n)$.

**Proof.** Let $(T, W)$ be an tree decomposition of width $k = tw(G)$, and root $T$ at some arbitrary vertex. Now for any vertex $i$ of $T$, define $G_i$ the subgraph of $G$ induced by vertices that belong to bags $W_i$, where $i'$ is a descendent of $i$. Now for any stable set $S$ which is a subset of $W_i$, define

$$\phi(S, t) = max(|S'|, S' \text{ stable set of } G_i \text{ such that } S' \cap W_i = S).$$

If $t$ is a leaf of $T$ then $\phi(S, t) = |S|$. Assume now that $t$ has children $t_1, \ldots, t_p$, and assume the value $\phi(S, t_i)$ is known for any $i$ and stable set inside $W_i$. We prove that we can compute $\phi(S, i)$ for any stable set $S \subseteq W_i$. The important and easy fact is that a set $S'$ is a stable set of $G_i$ if and only if its intersection with $W_i$ is a stable set, as well as all its intersections with the graphs $G_i$. Therefore for any $i$ let $S_i$ be a stable set of $W_i$ such that $S \cap W_i = S_i \cap W_i$ and $\phi(S_i, t_i)$ maximal. To find such a set we need to explore all stable sets of $W_i$, which takes time $O(2^k)$ for every $i$. Moreover we have that

$$\phi(S, t) = |S| + \sum_i (\phi(S_i, t_i) - |S_i \cap W_i|).$$

The total computing time is therefore $\sum_{e \in V(T)} 2^k 2^k p_i$ where $p_i$ is the number of children of node $i$. By using a tree decomposition with a linear number of nodes we get a complexity of $O(4^k k^2 . n)$.

**Theorem 5.2**

*Decide if $G$ is 3 colourable can be solved in linear time.*

**Proof.** Let $X_i$ be any bag of $T$, and fix it as a root of $T$. Denote by $T_i$ the trees rooted at vertex $X_i$. For each $i$ denote by $C_i$ the set of all 3 colourations of $G_{|X_i}$ that can be extended to 3 colourations of $T_i$. If we manage to compute all $C_i$, then we conclude by checking if $C_i$ is non empty.

For every leaf $X_i$ computing $C_i$ can be done exhaustively in time $O(3^i)$. For every internal node $X_i$ whose children have been dealt with, we compute all 3 colourations of $G_{|X_i}$ to get a set $C_i$. Then for each colouration $c$ in this set, we keep it in $C_i$ if for every children $X_j$, there exists a coloration $c_j$ such that it coincides with $c$ when restricted to $X_i \cap X_j$. It takes time $3^i 3^i p_i r_i^2$ if $p_i$ is the number of children of $X_i$.

In total this gives $\sum X_i 9^i r_i^2 (p_i + 1) = O(9^i r_i^2 . n)$.

Often it is useful to design those algorithms to work with a tree decomposition that has nice structural properties. We define those below.

**Definition 5.3**

A **nice tree decomposition** of $G$ is a tree decomposition where $T$ is a **rooted binary tree with**
bags \((W_t)_{t \in V(T)}\) and each inner node \(t\) is of three possible kind:

- **Join**: \(t\) has two children \(t_1\) and \(t_2\) and \(W_t = W_{t_1} = W_{t_2}\).
- **Introduce**: \(t\) has one child \(t'\) and \(W_t = W_{t'} \cup \{x\}\) where \(x \notin W_{t'}\)
- **Forget**: \(t\) has one child \(t'\) and \(W_t = W_{t'} \cup \{x\}\) where \(x \notin W_t\)

The important fact is that for every graph one can find an optimal tree decomposition that is simple.

**Proposition 5.4**

Given a tree decomposition of width \(w\) and with \(n\) nodes, it is possible in time \(O(w^{O(1)}n)\) to transform it into a simple one with width \(w\) and with \(O(nw)\) nodes.

**Proof.** Root the decomposition arbitrarily. For each internal node with \(p\) children, it is possible to add \(2p\) new join nodes to make it binary. Then for each node \(t\) with one child \(t'\). It possible now to replace the edge \(tt'\) by a path of at most \(2w\) forget or introduce nodes. \(\square\)

Using this kind of special decomposition often makes it easier to design dynamical programming algorithms using tree decompositions. See Exercise ??

Here is a list of results one can prove similarly using a tree decomposition of treewidth \(k\).

**Theorem 5.5**

Let \(G\) be given with a tree decomposition of width at most \(k\).

1. Computing \(\chi(G)\) can be done in time \(O(f(k)n)\)
2. Computing \(\omega(G)\) can be done in time \(O(2^{k}k^{k}n)\)
3. Computing \(\gamma(G) := \min \{|X|, X \text{ dominating set }\}\), can be done in time \(O(f(k)n)\).
4. Deciding if \(G\) has a hamilton cycle can be done in time \(O(f(k)n)\)

### 5.3 Monadic Second Order Logic

A celebrated algorithmic meta-theorem of Courcelle ([9]) generalises all the previous results to monadic second order formulas. Let us explain briefly what these are. We are interested into logical formulas on the graph seen as a relational structure. This means variables are vertices or edges, and one can build formulas inductively using:

- atomic formulas: \(x = y, v \in X, e \in F\) for subsets of vertices or edges.
- the binary relation \(xIe\) which is satisfied if \(x \in V\) and \(x\) is incident with \(e \in E\).
- logical operators \(\lor\) and \(\land\) and \(\neg\)
- quantifiers \(\forall\) and \(\exists\)
To be precise about quantifiers, first order formulas is the fragment of logic where one is allowed to use quantifiers over vertices and edges (\(\forall v \phi(v)\)), MSO\(_1\) is the fragment where one is allowed in addition to quantify over sets of vertices, and eventually MSO\(_2\) is a larger fragment where in addition one can quantify over sets of edges.

Every time one property is written as such a formula, one can give it a name and use it as a macro to produce other formulas (the length of the formula of course should take this into account). For example \(x \neq y\) instead of \(\neg(x = y)\) or \(X \subseteq Y\) instead of \(\forall x \in V(x \in X \Rightarrow x \in Y)\).

In a formula, a variable that is not in the scope of a quantifier is called a free variable. For example the property \(adj(x, y)\) that vertices \(x\) and \(y\) are adjacent has two free variables and can be written

\[(x \in V) \land (y \in V) \land (x \neq y) \land (\exists e \in ExIe \land yIe)\]

and the fact that the graph contains a vertex adjacent to all others is formulated by

\[\exists x \in V(\forall y \in V (y \neq x) \Rightarrow adj(x, y))\]

These were first order formulas, with second order one can express for example 3 colourability:

\[
Col3 := \exists X_1 \subseteq V \exists X_2 \subseteq V \exists X_3 \subseteq V
       \quad (\forall x \in V \quad (x \in X_1 \lor x \in X_2 \lor x \in X_3)
       \quad \land \neg(x \in X_1 \lor x \in X_2) \land \neg(x \in X_1 \land x \in X_3) \land \neg(x \in X_2 \land x \in X_3))
       \land (\forall xy \in E \quad \neg(x \in X_1 \land y \in X_1) \land \neg(x \in X_2 \land y \in X_2) \land \neg(x \in X_3 \land y \in X_3))
\]

The theorem of Courcelle asserts that every such property is easy to decide for bounded treewidth graphs.

**Theorem 5.6**

Let \(\phi\) be a MSO\(_2\) formula of length \(k\), and let \(G\) be a graph on \(n\) vertices with \(tw(G) \leq t\). There exists an algorithm that performs in time \(f(k, t).O(n)\) to decide whether \(G\) satisfies \(\phi\).

Note that the formula that says that \(G\) has a vertex cover of size at most \(k\) has length \(O(k)\).

\[\exists x_1 \exists x_2 \ldots \exists x_k \forall e \in E \quad (x_1Ie \lor x_2Ie \lor \ldots \lor x_kIe)\]

So applying the previous theorem would only give a \(f(k, w).O(n)\) algorithm to decide if a graph of treewidth at most \(t\) has a vertex cover of size at most \(k\).

There is a version of Courcelle Theorem for optimisation problems instead of decision ones.

**Theorem 5.7**

Let \(\phi(S)\) be a MSO\(_2\) formula of length \(k\) with free variable \(S\), and let \(G\) be a graph on \(n\) vertices with \(tw(G) \leq t\). There exists an algorithm that performs in time \(f(k, t).O(n)\) that can find the minimum size of an \(S\) that satisfies \(\phi(S)\).

(For Vertex Cover the formula is then just \(\forall e \in E \exists x \in S \ xIe\).)

This is an example of such a theorem, more complicated versions exist (with several free variables for example).

Courcelle’s Theorem is a powerful tool for classification but has limitations because there are no good estimates on the running time. So in order to get practical algorithms, one still needs to design algorithms for each problem.


## 5.4 Computing tree width

Until now, we have delayed the problem of finding a small width decomposition. The main problem with computing treewidth is that the following problem is NP-complete ([5]):

**Input**: $G, t$

**Output**: TRUE if $\text{tw}(G) \leq t$

(It is interesting to mention here that it is still open whether the same problem is polynomial for planar graphs.)

On the other hand, since we are interested into using tree decompositions to devise FPT algorithms with parameter $\text{tw}(G)$, we would be happy to compute such a tree decomposition in time $O(f(\text{tw}(G)).P(n))$. There exists such algorithms (even linear in $n$, see [6] for a $O(2^{\text{tw}(G)^2}n)$ algorithm), but in the sole purpose of having FPT algorithms, we need in fact only something weaker, such as the following.

**Theorem 5.8**

There exists an algorithm with input a graph $G$ and an integer $k$ and that outputs in time $O(f(k).n^2)$:

- either $\text{tw}(G) \geq k$
- or a tree decomposition of width at most $4k - 1$.

Note that this is enough to apply the algorithms described in the previous section for bounded treewidth graphs. Indeed by applying the previous theorem for $k = 1, k = 2, k = 3, ...$ one is guaranteed to find a tree decomposition of $G$ of width at most $4\text{tw}(G)$ in time $O(f(\text{tw}(G)).n^2)$.

Before proving the theorem let us define the notion of **good separator** for a set $W$ of vertices.

$S$ is a good separator for $W$ if $S$ disconnects $G$ into non-trivial subsets $V_1$ and $V_2$ such that for $i = 1, 2$, $V_i$ contains at most $2|W|/3$ vertices of $W$ (and therefore at least $|W|/3$ - these are often called $1/3$-$2/3$ separators). One can prove that if $\text{tw}(G) < k$, every $X$ of size at least $2k + 1$ admits a good separator of size at most $k$ (Exercise 5.4). The main ingredient for the proof is the fact that the converse is almost true.

**Proof.**

We prove inductively that there exists an algorithm for the following (it suffices to apply it with $W = \emptyset$ to get the theorem).

**Input**: $G$, $W \subseteq V(G)$ such that $|W| \leq 3k$

**Output**: A certificate that $\text{tw}(G) \geq k$ or a rooted tree decomposition $T$ of $G$ of width at most $4k - 1$ where $W \subseteq \text{root}(G)$

If $G$ has less than $4k$ vertices then put all vertices in a single bag.

If not, but $W$ has less than $2k + 1$ vertices, then augment $W$ arbitrarily by adding vertices until its size is at least $2k + 1$. If now $W$ admits no good separator of size at most $k$, than by what precedes it is a certificate than $\text{tw}(G) > k$. Assume for the moment that it exists and we are able to compute it. We are given $S$ such that $G \setminus S$ is the disjoint union of $G_1$ and $G_2$ with $W \cap V(G_i) \leq 2|W|/3$. Define $W_i = S \cup (W \cap V(G_i))$. Then $|W_i| \leq k + 2\frac{2}{3}k = 3k$ and we can inductively apply the algorithm on $(G_i, W_i)$ for $i = 1, 2$ to get either a certificate that $\text{tw}(G) \geq k$ or two decompositions $T_1, T_2$ of $G_1$ and $G_2$. It
suffices to add a root bag containing all vertices in \( W \cup S \) attached to the roots of \( T_1 \) and \( T_2 \) to get the desired tree decomposition (note that \( W \cup S - 1 \leq 4k - 1 \)).

The only thing remaining is to prove that we have an algorithm to find a good separator for \( W \) one exists. Such a set exists if and only if one can partition \( W \) into three subsets \( W_1, W_2, W_0 \) such that \( W_1 \) and \( W_2 \) have size at most \( 2|W|/3 \), and \( W_0 \) is a subset of a separator of size at most \( k \) separating \( V_1 \) and \( V_2 \) where \( W_i \subset V_i \). Observe that \( W_0 \) can be extended into such a separator if and only if in \( G \setminus W_0 \), there are at most \( k - |W_0| \) disjoint paths from \( W_1 \) to \( W_2 \). This can be tested by a max flow technique in time \( O(k^2n) \) (because the graph has at most \( kn \) edges, otherwise it cannot have treewidth at most \( k \)). If the answer is no for every partition there the set \( W \) does not have a good separator. Otherwise we find the separator. There are less than \( 3^k \) ways of defining the partition \( W_0, W_1, W_2 \) so this gives complexity \( O(27^k, k^2n^2) \) in total since the tree decomposition has at most \( n \) nodes.

\[ \square \]

As mentioned before, the proof above contains the following result

**Theorem 5.9**

Let \( G \) be a graph such that every \( X \subset V(G) \) of size at least \( 2k + 1 \) admits a good separator of size at most \( k \), then \( \text{tw}(G) \leq 4k - 1 \).

which gives a proof of the approximation of the duality theorem since if \( X \) has size at least \( 2k + 1 \) the set of connected subgraphs containing at least \( k + 1 \) vertices of \( X \) forms a bramble.

**Theorem 5.10**

For any graph \( G \)

\[ \text{tw}(G) \leq 4bn(G) - 1 \]

### 5.5 FPT Algorithms

A parametrized algorithmic problem is a problem where a certain parameter is given in addition to the input. We usually denote by \( k \) the parameter and \( n \) the size of the input. There are roughly three possibilities for a parametrized algorithmic problem.

- Either the problem is already hard for fixed \( k \).
  Example = compute \( \chi(G) \) with parameter the solution. NP hard for \( k = 3 \).

- Or the problem is polynomial of \( k \) fixed.
  Example = Decide if \( \alpha(G) \leq k \) with parameter \( k \) needs exhaustive search : \( O(n^k) \).

- FPT : Algorithm in time \( O(f(k)n^{O(1)}) \)

The parameter \( k \) can be directly the size of the solution, or an implicit parameter of the input graph (like the diameter, maximum degree, the treewidth).

We have seen in the previous sections several examples of problems that are FPT when the parameter is treewidth.

In this section we will show techniques to design FPT algorithms (including some based on treewidth, but not only). We illustrate through the analysis of the classic example of \( k \)-**Vertex**
Cover Problem \((k\text{-VC})\). The parameter will be the size of the solution \(k\). Even though a vertex cover is the complement of an independent set, note that a FPT algorithm parametrized by the size of the solution for the first problem does not give one for the second (in fact no \(n^{o(k)}\) is known for max independent set).

5.5.1 Existence of FPT via Robertson and Seymour Theorems

Assume \(C\) is a class of graphs, and let us look at the following algorithmic question: Can we find a \(f(k) \text{Poly}(n)\)-time algorithm to decide if in a graph \(G\) on \(n\) vertices, there exists \(X \subseteq V(G)\) with \(|X| \leq k\) and \(G \setminus X \in C\)? Note that if the class \(C\) is minor closed, then the answer is YES, there exists such an algorithm, since the set of YES instances for this problem is still minor closed (why?). Hence by Robertson and Seymour theorems, there exists a finite list of obstructions, and thus a \(O(n^3)\) algorithm (where the constants depends on \(k\)).

Note that \(k\)-Vertex Cover corresponds exactly to this problem for being the class of stable sets. Similarly, if we take \(C\) to be the set of forests, we get the classical problem of deciding whether it is possible to intersect all cycles of a graph with less than \(k\)-vertices (\(k\)-Feedback Vertex Set Problem).

The problem with this answer is that it is just existential since we do not know how to get the bounds (Robertson and Seymour theorems are not constructive).

5.5.2 TreeWidth Based Algorithms

How can we use the fact that we have good algorithms for bounded treewidth graphs (either by Courcelle’s Theorem, or by dynamical programming if we seek practical ones) to design algorithms for all graphs?

Let us take the example again of \(k\)-Vertex Cover. Remember that for a graph \(G\) we are able in time:

- \(f_1(k) \text{O}(n)\) to decide whether \(\text{tw}(G) \leq k\) and find a tree decomposition of width \(k\) in that case.
- \(f_2(k) \text{O}(n)\) to decide \(G\) has a \(k\)-VC if we are given a tree decomposition of width \(k\) of \(G\).

The missing part is in fact trivial: for any graph \(G\), if \(G\) has treewidth larger than \(k\), then it has no \(k\)-VC. (why?)

This gives a scheme for designing FPT algorithms as long as the problem we are solving has some property that implies that there is a bound \(t\) such that if \(\text{tw}(G) > t\) then either the answer is always YES or always NO.

5.5.3 Branching

Let us prove an easy \(2^k \cdot \text{O}(n)\) algorithm for \(k\)-VC by a technique called branching. Consider an edge \(xy\), by definition any solution must contain either \(x\) or \(y\). It is then easy to prove that \(G\) has a \(k\)-VC if and only if \(G \setminus x\) has a \((k - 1)\)-VC or \(G \setminus y\) has a \((k - 1)\)-VC, which implies the desired complexity.

This can be improved by other rules of branching or reduction.

- if every vertex has degree at most 2, then the problem can be solved in polynomial time.
if \( x \) is a vertex of degree \( d \geq 3 \), then either the solution contains \( x \), or it has to contain all of its neighbors. So \((G,k)\) is replaced two branches with input/parameter size \((n-1,k-1)\) and \((n-d-1,k-d)\).

The maximum number of leaves \( l(k) \) of a tree constructed this way satisfies \( l(k) \leq l(k-1)+l(k-3) \). We can prove that \( l(k) \leq c^k \). It means that \( c \) satisfies \( c^3 - c^2 - 1 \geq 0 \), and we are looking for a smallest root. Such an equation has a unique positive root \( \approx 1.47 \).

### 5.5.4 Kernelization

The idea of kernelization is to do the a preprocessing which transforms an instance \((G,k)\) in an instance \((G',k')\), called a kernel, such that

- \((G,k)\) is a YES-instance if and only if \((G',k')\) is a YES-instance
- \( k' \leq k \)
- the size \( G' \) is bounded by a function of \( k \).

It is sufficient then to apply a brute force algorithm, to get an FPT algorithm.

For \( k - VC \), a way to construct a kernel is by applying the following observations to an input graph \( G \). Assume \( G \) has a \( k \)-VC \( S \):

- If \( v \) is an isolated vertex, it is not in the solution.
- Every vertex of degree larger than \( k \) must be in \( S \).
- If every vertex has degree at most \( k \) then the total number of edges does not exceed \( k^2 \), and the total number of vertices cannot exceed \( k^2 + k \).

Using this one can construct a kernel of quadratic size, and by brute force one gets a \( O(k^2 + n + m) \) algorithm. Again by using more elaborated techniques, one can find better kernels and thus better algorithms (\( 2k - c \log(k) \) is the best known kernelization).

### 5.5.5 Iterative Compression

This technique is a very powerful one, many of the difficult FPT algorithms are based on it. Here we continue to illustrate it via the example of vertex cover, even though in this case the technique does not yield a better algorithm than the other principles exposed before.

The central idea here is to do the following : instead of finding directly in FPT time a solution of type \( k \), we prove a compression result, i.e. an algorithm that given a solution \( S \) of size \((k+1)\) will either find a solution of size \( k \), or certify that non exist.

For every \( S' \subset S \), let \( S'' = N(S \setminus S') \setminus S \). Then if \( S \setminus S' \) contains no edge, \( T = S' \cup S'' \) is a vertex cover. If for some \( S' \) \( T \) has size at most \( k \) then a \( k \)-vertex cover exists, otherwise no, since any \( k \)-VC must be of this kind.

Now if \( V(G) = \{v_1, ..., v_n\} \), and \( G_i = G[v_1, ..., v_i] \) for all \( i \in \{1, ..., n\} \), define

- \( S_1 = \emptyset \)
- If \( S_i \) is a VC for \( G_{i+1} \) then \( S_{i+1} := S_i \), otherwise else \( S_{i+1} = S_i \cup v_{i+1} \)
- If \( |S_i| > k \) use the previous compression algorithm to reduce its size by 1 or output NO.
Here this will give a $O(2^kn)$ algorithm.
This technique will work for any minimisation problem for which we are able to construct a sequence $G_1, G_2, ..., G_{n=G}$ and

- a $k$-solution for $G_1$ exists.
- a $k$-solution for $G_i$ can be extended (in poly time) to a $(k + 1)$-solution of $G_{i+1}$
- if $G_{i+1}$ has a $k$ solution then $G_i$ has one
- if a $(k + 1)$-solution of $G_{i+1}$ is given, there is an FPT algorithm to decide if it has a $k$-solution.

5.5.6 Colour Coding (NON TRAITÉ CETTE ANNEE)

Color coding is a method proposed in 1994 by Alon, Yuster, and Zwick (see []). For this paragraph, we change the parametrized problem to the one of deciding whether a graph contains a path on $k$-vertices, $k$ being the parameter. Of course this in NP-complete since $k = n$ solves hamiltonicity.

The surprising idea is to transform the problem into the following: assume every vertex is randomly given a color from $\{1, 2, ..., k\}$. Now the problem is to decide in FPT time whether $G$ contains a path on $k$ vertices using all $k$-colours.

Assume first we can do that. Now if a $k$-path $P$ exists, what is the probability that it receives all $k$ colours? The answer is $k! / k^k$. Using Stirling’s formula ($k! \sim \sqrt{2\pi k} (\frac{k}{e})^k$) we get that the probability is at least $1/e^k$. Now it means that if we repeat the argument $N$ times, the probability that $P$ is never colorful is $(1 - e^{-k})^N$ which converges to 0.

Two questions to solve:

- How to solve the coloured version?
- How to derandomize?

To solve the coloured version, first note that we can slightly transform it into a rooted version: we want the path to start in a specified vertex $s$ (it is enough to add a universal vertex with a new colour). By dynamical programming one can compute the function $f(x, C)$ which is equal to 1 if there exists a colourful path from $s$ to $x$ and 0 otherwise. Start from $x = s$ $C = \text{col}(s)$ and then go in BFS fashion. It takes time $2^{k \cdot n}$.

To derandomize, one can use the notion of $k$-perfect family of hash functions. It is a family of functions from a set $V$ to $\{1, \ldots, k\}$ such that for any $K \subset V$ of size $k$, there exists one function in the family such that $f$ is bijective when restricted to $K$.

Several constructions exist, a theorem of Siegel et al (see []) proves for example the existence of a family of $k$-perfect has functions for a set of size $n$ of cardinality $2^{O(k) \log^2(n)}$, which can be computed in time $2^{O(k)n \log^2(n)}$.

5.6 Exercises

Exercise 5.1. Prove that given a nice tree decomposition of width $k$ one can obtain an algorithm in $2^kO(n)$ for max independent set.

49
Exercise 5.2. Prove that given a nice tree decomposition of width $k$ one can obtain an algorithm in $k^2O(n)$ for min Feedback Vertex Set (minimum set whose removal leaves the graph acyclic).

Exercise 5.3. Prove that the existence of a hamiltonian cycle can be expressed as an $MSO_2$ formula. Same for the fact that a fixed graph $H$ is a minor of the input graph.

Exercise 5.4. Using a proof similar to the proof of Proposition 4.22, prove that if $tw(G) \leq k$, then for any $W \subseteq V(G)$ of size at least $2k + 1$, there exists a $S$ of size at most $k + 1$ that is good separator for $W$, i.e. the removal of $S$ disconnects $G$ into non-trivial subsets $V_1$ and $V_2$ such that $V_1$ and $V_2$ both contain at most $2|W|/3$ vertices of $W$.

Exercise 5.5. Consider the following algorithmic problem:

$k$-LEAVES SPANNING TREE
Input : $G$ a $n$-vertex connected graph.
Output : YES if $G$ admits a spanning tree with at least $k$ leaves, and NO otherwise.

1. Why does there exist an $f(k)O(n^3)$ algorithm for this problem?
2. Prove that $G$ is a YES-instance if and only if $G$ is connected and has $K_{1,k}$ as a minor.
3. Using the bramble-treewidth duality theorem, prove that if $G$ does not contain $K_{1,k}$ as a minor, then $tw(G) \leq k - 1$.
4. Prove now a $f(k)O(n^2)$ algorithm for this problem.

In the next two questions we are going to prove the bound on treewidth for $K_{1,k}$ minor free graphs without using the bramble treewidth duality theorem. To do this, let us consider a depth-first search tree $T$ of the graph.

5. Prove that any edge of $G$ goes from a vertex to one of its ancestors in $T$.
6. Use this tree $T$ to define a tree decomposition of width at most $r - 1$ of $G$.

Exercise 5.6. Consider the following algorithmic problem:

SUBGRAPH ISOMORPHISM.
Input : two graphs $H$ and $G$.
Output : decide if $G$ has a subgraph isomorphic to $H$.

Denote $n = |V(G)|$, $k = |V(H)|$ and $w = tw(G)$.

1. Why does there exist a FPT algorithm in time $f(k, w)O(n)$ to solve this problem?

The goal of the following is to prove that in the case where $G$ is planar, one can suppress the dependency on the treewidth to get an FPT algorithm only for parameter $k$.

For this we need a definition: a planar graph is said to be outerplanar (or 1-outerplanar) if there exists a planar embedding of $G$ where all its vertices lie on the outer face (for example, $K_4$ is planar but not outerplanar). A embedding of a graph is $k$-outerplanar if by deleting its outer face, one gets an embedding that is $k - 1$-outerplanar graph. A graph is $k$-outerplanar if has a $k$-outerplanar embedding. It can be proven (see Exercise 4.19) that $k$-outerplanar graphs have treewidth at most $3k$.

2. Let $G$ be a planar graph with a planar embedding, let $L_1$ be its outer face and for $i \geq 1$, $L_i$ be the outer face of $G \setminus \bigcup_{j<i} L_j$. For $0 \leq s \leq k$, let $G_s$ obtained by deleting all layers $L_i$ with $i \equiv s$
mod \([k + 1]\). Prove that one can decide in time \(f(k)n\) if \(G\) has a subgraph isomorphic to \(H\) and conclude.
Bibliography


