CHAPTER 2

Tree-Decompositions of Graphs

Tree-decompositions of graphs play an important role in graph structure theory, in the theory of algorithms and in practical computation.

2.1. Introduction to tree-Width

Definition 2.1.1. Let G be a graph. A *tree-decomposition* of G is a pair (T, W), where T is a tree and $W = (W_t : t \in V(T))$ is such that

(i) $\bigcup_{t \in V(T)} W_t = V(G)$, and every edge of G has both ends in some W_t, and

(ii) if $t, t', t'' \in V(T)$ and t' lies on the path from t to t'' in T, then $W_t \cap W_{t''} \subseteq W_{t'}$.

The *width* of (T, W) is $\max\{|W_t| - 1 : t \in V(T)\}$, and the *tree-width* of G is the minimum width of a tree-decomposition of G.

Exercise 2.1.2 ([**10**])**.** Let (T, W) be a tree-decomposition of G, let H be a connected subgraph of G, and let $t, t'' \in V(T)$ be such that $V(H) \cap W_t \neq \emptyset \neq V(H) \cap W_{t''}$. Prove that if $t' \in V(T)$ lies on the path of T between t and t'', then $V(H) \cap W_{t'} \neq \emptyset$.

Exercise 2.1.3 ([**10**])**.** Let (T, W) be a tree-decomposition of G, and let $e \in E(T)$ with ends t_1, t_2. Then $T \backslash e$ has exactly two components T_1, T_2 say. Prove that there is no edge between $\bigcup_{t \in V(T_1)} W_t - (W_{t_1} \cap W_{t_2})$ and $\bigcup_{t \in V(T_2)} W_t - (W_{t_1} \cap W_{t_2})$.

Exercise 2.1.4. Let (T, W) be a tree-decomposition of a graph G, and let H be a complete subgraph of G. Prove that there exists $t \in V(T)$ such that $V(H) \subseteq W_t$. Deduce that the complete graph K_n has tree-width $n - 1$.

Exercise 2.1.5. Let (T, W) be a tree-decomposition of G, and let e be an edge of G with ends u and v. Let $w \notin V(G)$ and for $t \in V(T)$ let $W'_t = (W_t - \{u, v\}) \cup \{w\}$. Prove that $W/e := (W'_t : t \in V(T))$ is a tree-decomposition of G/e, the graph obtained from G by contracting e. Deduce that if G has an H minor, then the tree-width of H is at most the tree-width of G.

Exercise 2.1.6. Prove that a simple graph G has tree-width at most 1 if and only if G is a forest.

Exercise 2.1.7. Prove that a graph G has tree-width at most 2 if and only if G is series-parallel.

Exercise 2.1.8 ([**8**])**.** A graph G is chordal if and only if G has a tree-decomposition (T, W) such that every W_t is a clique.

Exercise 2.1.9. If G is a graph and $k \geq 0$ is an integer, then G has tree-width at most k if and only if G is a subgraph of a chordal graph with no subgraph isomorphic to K_{k+2}.

Exercise 2.1.10. Prove that every graph G of tree-width t has a set $X \subseteq V(G)$ of size at most t such that every component of $G \backslash X$ has at most $|V(G)|/2$ vertices.

Exercise 2.1.11. Prove that every graph G of tree-width t on n vertices has a separation (A, B) of order at most t such that both $A - B$ and $B - A$ have at most $2n/3$ vertices.

Exercise 2.1.12. Let $n \geq 2$ be an integer. Prove that the $n \times n$ grid (the adjacency graph of the $n \times n$ chessboard) has tree-width n.

Exercise 2.1.13 ([**13**])**.** For every planar graph H there exists an integer n such that the $n \times n$ grid has a minor isomorphic to H.

Theorem 2.1.14. *For every planar graph H there exists an integer k such that if a graph G has tree-width at least k, then it has an H minor.*

Exercise 2.1.15. If H is non-planar then no such integer exists.

Remark 2.1.16. We will prove Theorem 2.1.14 later. The above theorem was first proved in [**11**]. A simpler proof with a better bound on k was given in [**13**]; the bound there is $k = 20^{2n^5}$, where $n = 2|V(H)| + 4|E(H)|$. A yet simpler proof with only a marginally worse bound was given in [**6**]. It seems to be an interesting and difficult problem to decide if k is bounded by some polynomial in n.

Exercise 2.1.17. Let H be the $n \times n$ grid, and let k be as in the above theorem. Prove that $k \geq \Omega(n^2 \log n)$.

Hint. Random graphs.

Definition 2.1.18. Consider the following cops-and-robbers game. The robber stands on a vertex of the graph and can at any time run at great speed to any other vertex along a path of the graph. He is not permitted to run through a cop, however. There are k cops, each of whom at any time either stands on a vertex or is in a helicopter (that is, is temporarily removed from the game). The objective of the player controlling the movement of the cops is to land a cop via helicopter on the vertex occupied by the robber, and the robber's objective is to elude capture. (The point of the helicopters is that cops are not constrained to move along paths of the graph—they move from vertex to vertex arbitrarily.) The robber can see the helicopter approaching its landing spot and may run to a new vertex before the helicopter actually lands. Thus, for the cops to capture the robber they will need to first occupy all vertices adjacent to the vertex where the capture is to take place, because otherwise the robber will be able to run to a different vertex and not be captured.

There are two forms of this game of interest. In the first, the robber is invisible, and so to capture him the cops must methodically search the whole graph. (An equivalent problem is that of clearing the vertices of some plague which infects along edges.) In the second form of the game the cops can see the robber at all times—the difficulty is just to corner him somewhere.

Exercise 2.1.19. Prove that if G has tree-width at most k, then $k + 1$ cops can capture a visible robber in G.

Definition 2.1.20 ([**15**])**.** Let G be a graph and let $k \geq 0$ be an integer. If $X \subseteq V(G)$ then by an *X-flap* we mean the vertex-set of a component of $G \backslash X$.

An *escape* in G of order k is a function σ which assigns, to every $X \subseteq V(G)$ with $|X| < k$, a set $\sigma(X) \subseteq V(G)$ such that for every $X, Y \subseteq V(G)$ with $|X|, |Y| < k$

(i) $\sigma(X)$ is a union of X-flaps of G,

(ii) $\sigma(X) \neq \emptyset$, and

(iii) if $X \subseteq Y$ then $\sigma(Y)$ is the union of precisely those Y-flaps that intersect $\sigma(X)$.

Escapes correspond to winning strategies for the robber.

Exercise 2.1.21. A graph G has an escape of order k if and only if $k-1$ cops cannot capture a visible robber.

Definition 2.1.22. Let G be a graph. We say that two set $X, Y \subseteq V(G)$ *touch* if either $X \cap Y \neq \emptyset$, or $X \sim Y$ in G.

Definition 2.1.23 ([**15**])**.** Let G be a graph, and let $k \geq 0$ be an integer. A *haven* of order k in G is a function assigning to every set $X \subseteq V(G)$ with $|X| < k$ an X-flap $\beta(X)$ such that if $X \subseteq Y \subseteq V(G)$ and $|Y| < k$, then $\beta(Y) \subseteq \beta(X)$. A *convex haven* of order k in G is a function assigning to every set $X \subseteq V(G)$ with $|X| < k$ an X-flap $\beta(X)$ such that if $X, Y \subseteq V(G)$ and $|X|, |Y| < k$, then $\beta(X)$ and $\beta(Y)$ touch.

Exercise 2.1.24. Every haven is an escape. Every convex haven is a haven.

Definition 2.1.25. A *bramble* in a graph G is a set $\mathcal{B}$ of subsets of $V(G)$ that pairwise touch. The *order* is the minimum cardinality of a set that intersects every member of $\mathcal{B}$.

Exercise 2.1.26. A graph has a bramble of order k if and only if it has a convex haven of order k.

Theorem 2.1.27 (Min-max theorem for tree-width [**15**])**.** *Let $k \geq 0$ be an integer. A graph has a convex haven of order k if and only if it has tree-width at least $k-1$.*

Exercise 2.1.28. Prove only if.

Exercise 2.1.29. Let G be a graph, and let $k \geq 0$ be an integer. Then the following are equivalent.

(i) G has a bramble of order k,

(ii) G has a haven of order k,

(iii) $k-1$ cops cannot capture a visible robber in G,

(iv) $k-1$ cops cannot monotonely capture a visible robber in G, and

(v) G has tree-width at least $k-1$,

Remark 2.1.30. The decision problem whether the tree-width of G is at most k is obviously in NP. The min-max theorem comes close to proving that it is in co-NP, but not quite. In fact, it is not even clear whether a haven can be specified in polynomial space. Since computing tree-width is NP-hard, it is unlikely that it is in co-NP.

We now prove an asymptotic version of Theorem 2.1.27, which is much easier, and the proof technique is quite useful. We need a lemma.

Lemma 2.1.31. *Let G be a graph, let $k \geq 1$ be an integer, and let (A, B) be a separation in G of order at most $2k-1$. If $B - A \neq \emptyset$, then either*

(i) *G has a convex haven of order k, or*

(ii) *there exists a set $Z \subseteq B$ such that $|Z| \leq k$, $Z - A \neq \emptyset$ and every component of $G \backslash (A \cup Z)$ has at most $2k-1$ neighbors in $A \cup Z$.*

Proof. If for every set $Z \subseteq V(G)$ with $|Z| \leq k$ there exists a component of $G \backslash Z$ that contains at least k vertices of $A \cap B$, then let $\beta(Z)$ denote the vertex-set of such component. Since $|A \cap B| \leq 2k-1$ it follows that $\beta(Z)$ is well-defined. Furthermore, $\beta(Z) \cap \beta(Z')$ for every two sets $Z, Z' \subseteq V(G)$ of size at most $k-1$, and hence β is a convex haven of order k. Thus (i) holds.

We may therefore assume that there exists a set $Z_0 \subseteq V(G)$ of size at most $k-1$ such that every component of $G \backslash Z_0$ includes at most $k-1$ vertices of $A \cap B$. By adding a vertex of $B - A$ to Z_0 if necessary we obtain a set $Z_1 \subseteq V(G)$ of size at most k such that $Z_1 - A \neq \emptyset$ and every component of $G \backslash Z_1$ includes at most $k-1$ vertices of $A \cap B$. Let $Z = Z_1 \cap B$. It follows that every component of $G \backslash (A \cup Z)$ has at most $2k-1$ neighbors in $A \cup Z$, because it has at most $k-1$ neighbors in A and $|Z| \leq k$. Thus (ii) holds, as desired. □

We are now ready to prove an approximate version of Theorem 2.1.27.

Theorem 2.1.32. *Let G be a graph, and let $k \geq 1$ be an integer. Then either G has tree-width at most $3k-2$, or G has a convex haven of order k.*

Proof. Let (T, W) be a tree-decomposition of G such that

(1) $|W_t| \leq 3k-1$ for every $t \in V(T)$ of degree at least two,

(2) $|W_t \cap W_{t'}| \leq 2k-1$ for every edge $tt' \in E(T)$,

and subject to (1) and (2),

(3) $\bigcup W_t$ is maximum, where the union is taken over all $t \in V(T)$ such that $|W_t| \leq 3k-1$.

Such a choice is possible, because the unique tree-decomposition (T_0, W_0) of G with $|V(T_0)| = 1$ satisfies (1) and (2).

If (1) holds for every $t \in V(T)$, then (T, W) has width at most $3k-2$, and the theorem holds. Thus, we may assume that there exists a vertex $t_0 \in V(T)$ with $|W_{t_0}| \geq 3k$. By (1) the vertex t_0 has degree at most one. It follows from (3) that t_0 has degree exactly one, for otherwise (T, W) can be modified by adding a new vertex r and letting W_r be a nonempty subset of $V(G)$ of size at most $2k-1$, contrary to (3). Let t_1 be the unique neighbor of t_0, and let $X = W_{t_0} \cap W_{t_1}$. Then $|X| \leq 2k-1$ by (2). Let $A = V(G) - (W_{t_0} - X)$ and $B = W_{t_0}$. By Lemma 2.1.31 applied to the separation (A, B) we conclude that (i) or (ii) of that lemma holds. If (i) holds, then the theorem holds, and so we may assume that there exists a set $Z \subseteq B$ such that $|Z| \leq k$, $Z - A \neq \emptyset$ and every component of $G \backslash (A \cup Z)$ has at most $2k-1$ neighbors in $A \cup Z$. Let $\mathcal{C}$ be the set of components of $G \backslash (A \cup Z)$. We construct a new tree T' from T by adding, for each $J \in \mathcal{C}$, a new vertex t_J of degree one joined to t_0. We define

$$W'_t = \begin{cases} W_t & \text{if } t \in V(T) - \{t_0\} \\ X \cup Z & \text{if } t = t_0 \\ V(J) \cup N(J) & \text{if } t = t_J \text{ for } J \in \mathcal{C} \end{cases}$$

Then (T', W') is a tree-decomposition of G satisfying (1) and (2), contrary to (3), because Z includes at least one vertex of $V(G) - A = W_{t_0} - \bigcup_{t \neq t_0} W_t$. □

Lemma 2.1.33. *There exists a function f with the following property. Let G be a graph, let $k \geq 1$ be an integer, let $A, B, C, D \subseteq V(G)$, let $\mathcal{P}$ be a set of k^k disjoint A-B paths, and let $\mathcal{Q}$ be a set of $f(k)$ disjoint C-D paths. Then either*

(i) *G has a $k \times k$ grid mirror, or*

(ii) *there is a subset $\mathcal{Q}'$ of $\mathcal{Q}$ and a set $\mathcal{P}'$ of disjoint A-B paths in G such that $|\mathcal{P}'|$, $|\mathcal{Q}'| \geq k$ and every number of $\mathcal{P}'$ is disjoint from every member of $\mathcal{Q}'$.*

Proof. We assume that

($*$) for every edge $e \in E(G)$ that does not belong to any member of $\mathcal{Q}$ the graph $G \backslash e$ does not have k^k pairwise disjoint $A - B$ paths.

In particular, it follows that $|\mathcal{P}| = k^k$. If at least $f(k)/2$ of the paths $Q \in \mathcal{Q}$ have the property that Q is disjoint from at least k members of $\mathcal{P}$, then there is a subset $\mathcal{P}_1 \subseteq \mathcal{P}$ such that at least $f_1(k) := f(k)/\left(2\binom{k^k}{k}\right)$ paths in $\mathcal{Q}$ are disjoint from every member of $\mathcal{P}_1$. Thus (ii) of the lemma holds.

We may therefore assume that at least $f(k)/2$ of the paths $Q \in \mathcal{Q}$ intersect at least $k^k - k$ members of $\mathcal{P}$, and hence there exist $\mathcal{P}' \subseteq \mathcal{P}$ of size $k^k - k$ and $\mathcal{Q}' \subseteq \mathcal{Q}$ of size at least $f_1(k)$ such that every member of $\mathcal{P}'$ intersects every member of $\mathcal{Q}'$.

Let $P \in \mathcal{P}'$. Let $e_1, e_2, \ldots, e_r$ be edges on P, in order as P is transversed from A to B, not belonging to any member of $\mathcal{Q}$ and such that each component of $P \backslash \{e_1, e_2, \ldots, e_r\}$ intersects at least $2k^k + 1$ members of $\mathcal{Q}'$. With an appropriate choice of f we can find such edges, where r is as large as we wish as a function of k only.

By ($*$) there is, for each $i = 1, 2, \ldots, r$, a separation (A'_i, B'_i) of $G \backslash e_i$ of order $k^k - 1$ with $A \subseteq A_i$ and $B \subseteq B_i$. Then one end of e_i belongs to $A'_i - B'_i$ and the other to $B'_i - A'_i$. Let A_i consist of A'_i and the other end of e_i, and let $B_i = B'_i$. Then (A_i, B_i) is a separation of G of order k^k, and hence $A_i \cap B_i$ includes exactly one vertex from each member of $\mathcal{P}$.

Let $A_0 = A \cap V(\mathcal{P})$, $B_0 = V(G)$, $A_{r+1} = B \cap V(\mathcal{P})$ and $B_{r+1} = B \cap V(\mathcal{P})$. For $i = 0, 1, \ldots, r$ let $D_i := A_{i+1} \cap B_i$. Then $|\partial D_i| \leq 2k^k$, but $V(P) \cap D_i$ is intersected by at least $2k^k + 1$ members of $\mathcal{Q}'$. Thus at least one such member, say Q_i, satisfies $V(Q_i) \subseteq D_i$.

For $i = 0, 1, \ldots, r$ we define an auxiliary simple graph H_i with vertex-set $V(\mathcal{P}')$ by saying that $P_1 \sim P_2$ in H_i if some subpath of Q_i has one end in P_1, the other end in P_2 and is otherwise disjoint from all members of $\mathcal{P}'$. Then H_i is connected, because Q_i intersects every member of $\mathcal{P}'$. Given that r can be chosen arbitrarily large in terms of k, there is a set $I \subseteq \{0, 1, \ldots, r\}$ of size $k(k-1)$ such that the graphs H_i are the same for all $i \in I$; let H denote the common value. Here we mean the same graph, not just isomorphic. Since H is connected and has at least $k^k - k$ vertices, it follows that it either has a path on k vertices, or a vertex of degree at least $k+1$. In the former case it is easy to construct a $k \times k$ grid minor, and so we assume that $P_0 \in \mathcal{P}'$ is adjacent in H to $P_1, P_2, \ldots, P_k \in \mathcal{P}'$. Let $I = \{i_1, i_2, \ldots, i_{k(k-1)}\}$. In D_{i_1}, we find a path from P_1 to P_2 (that uses P_0 but is disjoint from $P_3, P_4, \ldots, P_k$), in D_{i_2} we find a path from P_2 to $P_3, \ldots$, and in $D_{i_{k-1}}$ we find a path from P_{k-1} to P_k. Thus if we think of $P_1, P_2, \ldots, P_k$ as the "rows" of our future grid, we have just constructed the first column. We then use the same process applied to $D_{i_{k+1}+1}, D_{i_k}, \ldots, D_{i_{2(k-1)}}$ to construct the second column, and so on. Thus G has a $k \times k$ grid minor, as desired. □

Proof of Theorem 2.1.14 By Exercise 2.1.13 and Theorem 2.1.32 it suffices to show that if G has a convex haven of order $(k^2+1)h(k)$, then it has a $k \times k$ grid minor, where the function $h(k)$ is defined as $f(f(\dots f(k)\dots))$; in other words, iterations of the function f from Lemma 2.1.33. We will assume, as we may, that $f(h) \geq k^k$. **TO BE COMPLETED**

We end this section with some motivational remarks. A tree-decomposition is really a collection of pairwise "non-crossing" separations. First let us recall the corresponding phenomenon for edge cuts. We say that two edge cuts $\delta(X)$ and $\delta(Y)$ in a connected graph G *cross* if all four of the sets $X \cap Y$, $X \cap Y^c$, $X^c \cap Y$ and $X^c \cap Y^c$ are non-empty. A family $\mathcal{C}$ of cuts is *laminar* if no two members of $\mathcal{C}$ cross.

Exercise 2.1.34. Let G be a connected graph, let T be a tree, and let $W = (W_t : t \in V(T))$ be a collection of disjoint subsets of $V(G)$ such that $W_t \neq \emptyset$ for every $t \in V(T)$ of degree one. For $e \in E(T)$ let T_1, T_2 be the two components of $T\backslash e$, and let C_e denote the cut $\delta\left(\bigcup_{t\in V(T_1)} W_t\right) = \delta\left(\bigcup_{t\in V(T_2)} W_t\right)$. Then $\mathcal{C} = \{\mathcal{C}_e : e \in E(T)\}$ is a laminar family of cuts in G, and every laminar family arises this way.

Two separations $(A_1, B_1), (A_2, B_2)$ of a graph G *cross* unless one of the following holds:

$$\begin{aligned} A_1 \subseteq A_2 &\quad \text{and} \quad B_2 \subseteq B_1 \\ A_1 \subseteq B_2 &\quad \text{and} \quad A_2 \subseteq B_1 \\ B_1 \subseteq A_2 &\quad \text{and} \quad B_2 \subseteq A_1 \\ B_1 \subseteq B_2 &\quad \text{and} \quad A_2 \subseteq A_1. \end{aligned}$$

A *dissection* of G is a set $\mathcal{D}$ of separations such that

(i) if $(A, B) \in \mathcal{D}$ then $(B, A) \in \mathcal{D}$
(ii) if $(A, B) \in \mathcal{D}$ then $A \neq V(G)$
(iii) if $(A_1, B_1), (A_2, B_2) \in \mathcal{D}$ and $A_1 \neq A_2$ then $B_1 \neq B_2$
(iv) no two members of $\mathcal{D}$ cross.

Let us observe the following useful lemma.

Lemma 2.1.35. *If $\mathcal{D}$ is a dissection, $(A_1, B_1), (A_2, B_2) \in \mathcal{D}$ and $A_1 \subseteq A_2$, then $B_2 \subseteq B_1$.*

Proof. Now $A_1 \not\subseteq B_1$ by (ii) above, and so $A_2 \not\subseteq B_1$. Similarly, $B_2 \not\subseteq A_2$ and so $B_2 \not\subseteq A_1$. If $A_2 \subseteq A_1$ then $A_1 = A_2$ and hence $B_1 = B_2$ by (iii) above, and so we may assume that $A_2 \not\subseteq A_1$. Since (A_1, B_1) and (A_2, B_2) do not cross, the desired conclusion follows. □

Let (T, W) be a tree-decomposition of a graph G. For $e \in E(T)$, let T_1, T_2 be the components of $T\backslash e$, and for $i = 1, 2$ let

$$A_i = \bigcup (W_t : t \in V(T_i)).$$

Then (A_1, A_2) is a separation of G, and we call (A_1, A_2) and (A_2, A_1) the separations *arising* from e. We say that a tree-decomposition (T, W) of a graph G is *proper* if

(i) for all $e \in E(T)$, if (A_1, A_2) arises from $e \in E(T)$ then $A_1 \neq V(G)$, and

(ii) for all $e, f \in E(T)$, if (A_1, A_2) and (B_1, B_2) arise from e and f respectively, and $A_1 = B_1$ then $A_2 = B_2$.

If (T, W) is proper, the set $\mathcal{D}$ of all separations arising from edges of T is clearly a dissection, which we call the *resultant dissection*. The following exercise provides the converse.

Exercise 2.1.36. For every dissection $\mathcal{D}$ of a graph G there exists a tree-decomposition (T, W) such that $\mathcal{D}$ is the resultant dissection.

2.2. Path-Width

Path-width is defined similarly as tree-width, except that the decomposition tree is constrained to be a path. Formally, it is more convenient to work with sequences rather than path, and so we define path-width as follows.

Definition 2.2.1 ([**9**])**.** A *path-decomposition* of a graph G is a sequence $W = (W_1, W_2, \ldots, W_l)$ such that

(i) $\bigcup_{i=1}^{l} W_i = V(G)$, and every edge of G has both ends in some W_i, and

(ii) if $1 \leq i \leq j \leq k \leq l$, then $W_i \cap W_k \subseteq W_j$. The *width* of W is $\max\{|W_i| - 1 : 1 \leq i \leq l\}$. The *path-width* of G is the least integer k such that G has a path-decomposition of width k.

The objective of this section is to prove the following:

Theorem 2.2.2. *Let G be a graph, and let F be a forest on n vertices. If G has path-width at least $n - 1$, then G has an F minor.*

Exercise 2.2.3. If F is not a forest, then no such bound exists.

Exercise 2.2.4. The bound $|V(F)| - 1$ is best possible.

Remark 2.2.5. Theorem 2.5.1 was originally proved by Robertson and Seymour in [**9**] with a much larger bound. The proof in [**3**] is easier. Diestel [**5**] found a yet simpler proof.

Exercise 2.2.6. Let T_k be the ternary tree of height k. Prove that the path-width of T_k tends to ∞ as $k \to \infty$.

Exercise 2.2.7. If G has an H minor, then the path-width of H is at most the path-width of G.

Definition 2.2.8. A graph G is an *interval graph* if for every $v \in V(G)$ there exists an interval $I_v \subseteq \mathbb{R}$ such that $u, v \in V(G)$ are adjacent in G if and only if $I_u \cap I_v \neq \emptyset$.

Exercise 2.2.9. Every interval graph is chordal.

Exercise 2.2.10. Find a chordal graph that is not an interval graph.

Exercise 2.2.11. Prove that a graph G is an interval graph if and only if the maximal cliques of G can be linearly ordered $C_1, C_2, \ldots, C_k$ so that if $v \in C_i \cap C_{i''}$ and $1 \leq i \leq i' \leq i'' \leq k$, then $v \in C_{i'}$.

Exercise 2.2.12. Let $k \geq 0$ be an integer, and let G be a graph. Prove that G has path-width at most k if and only if G is a subgraph of an interval graph that has no subgraph isomorphic to K_{k+2}.

Definition 2.2.13. Let G be a graph, let $n \geq 0$ be an integer. We denote by $\mathcal{S}_n$ the set of all sets $S \subseteq V(G)$ such that $G[S]$ has a path-decomposition $(W_1, W_2, \ldots, W_\ell)$ of width $< n$ such that $\partial S \subseteq W_\ell$.

Lemma 2.2.14. *Let G be a graph, let $n \geq 0$ be an integer, and let $\mathcal{S}$ be as in* Definition 2.2.13. *Then*

(1) $\emptyset \in \mathcal{S}$

(2) *if $S \in \mathcal{S}$, $S \subseteq S' \subseteq V(G)$, and $|\partial S| + |S' - S| \leq n$, then $S' \in \mathcal{S}$, and*

(3) *if $S \in \mathcal{S}$, $S' \subseteq S$ and G has $|\partial S'|$ disjoint paths from $\partial S'$ to ∂S, then $S' \in \mathcal{S}$.*

Proof. Condition (1) is clear. To prove (2) let $S \in \mathcal{S}$ and let $(W_1, W_2, \ldots, W_\ell)$ be a path-decomposition of $G[S]$ of width $< n$ with $\partial S \subseteq W_\ell$. If $S \subseteq S' \subseteq V(G)$ and $|\partial S| + |S' - S| \leq n$, then $(W_1, W_2, \ldots, W_\ell, \partial S, \partial S \cup (S' - S))$ is a path-decomposition of $G[S']$ of width $< n$ with $\partial S' \subseteq \partial S \cup (S' - S)$, as required. Thus (2) holds.

To prove (3) let S and S' be as stated, let $\partial S' = \{v_1, v_2, \ldots, v_k\}$, and let $P_1, P_2, \ldots, P_k$ be disjoint paths in G such that P_i has one end v_i and the other end in ∂S. Let $W = (W_1, W_2, \ldots, W_\ell)$ be a path-decomposition of $G[S]$ of width $< n$ with $\partial S \subseteq W_\ell$. Let $e_1, e_2, \ldots, e_t$ be the edges of $P_1 \cup P_2 \cup \cdots \cup P_k$; then the path-decomposition $W/e_1/e_2/\cdots/e_t$ obtained as in Exercise 2.1.5 gives rise to a desired path-decomposition of $G[S']$, showing that $S' \in \mathcal{S}$. □

Definition 2.2.15. Let G be a graph, let $n \geq 0$ be an integer, and let $\mathcal{S}$ be a set of subsets of $V(G)$. We say that $\mathcal{S}$ is a *stoppage in G of order $n+1$* if $\mathcal{S}$ satisfies (1)–(3) of the previous lemma. We will refer to (1), (2), and (3) as the first, second, and third stoppage axiom, respectively. We say that $\mathcal{S}$ is *proper* if $V(G) \notin \mathcal{S}$. We say that a set $S \in \mathcal{S}$ is *$\mathcal{S}$-free* if there is no set $S' \in \mathcal{S}$ with $S \subseteq S'$ and $|\partial S'| < |\partial S|$.

Thus if G has path-width at least n, then the set $\mathcal{S}$ defined in Definition 2.2.13 is a proper stoppage of order $n+1$. We will see soon that the converse also holds, but for now let us concentrate on the main result of this section.

Proof of Theorem 2.2.2. Let G be a graph, let F be a forest with vertex-set $\{v_1, v_2, \ldots, v_n\}$ numbered so that each v_i has at most one neighbor among $\{v_1, v_2, \ldots, v_{i-1}\}$, let G have path-width at least $n-1$, and let $\mathcal{S} = \mathcal{S}_{n-1}$, where $\mathcal{S}_{n-1}$ is as in Definition 2.2.13. By Lemma 2.2.14 the set $\mathcal{S}$ is a proper stoppage in G of order n.

Let us say that a set $S \in \mathcal{S}$ with $k := |\partial S| \leq n-1$ is *useful* if

(i) S is $\mathcal{S}$-free, and

there exist disjoint connected sets $X_1, X_2, \ldots, X_k$ such that

(ii) $|X_i \cap \partial S| = 1$, and

(iii) if $1 \leq i < j \leq k$ and $v_i \sim v_j$, then $X_i \sim X_j$.

Since $\emptyset$ is useful we may select a maximal useful set $S \in \mathcal{S}$. We now prove the following assertion.

($*$) If $S' \in \mathcal{S}$ and $|\partial S'| \leq |\partial S|$, then $S' = S$.

To prove this assertion let $S' \in \mathcal{S}$ satisfy $|\partial S'| \leq |\partial S|$. Then equality holds, because S is free. We claim that there exist k disjoint paths between ∂S and $\partial S'$. Indeed, otherwise G has a separation (A, B) of order $< k$ with $S \subseteq A$ and $\partial S' \subseteq B$. If we select (A, B) of minimum possible order, then there exist $|A \cap B|$ disjoint paths from $A \cap B$ to $\partial S'$, and hence $A \in \mathcal{S}$ by the third stoppage axiom, contrary to the freedom of S. This proves our claim that there exist k disjoint paths between ∂S and $\partial S'$. Let those paths be numbered $P_1, P_2, \dots, P_k$ so that P_i uses the vertex of $X_i \cap \partial S$. On letting $X_i' = X_i \cup V(P_i)$ we find that S' is useful, and hence the maximality of S implies that $S = S'$, as desired. This proves $(*)$.

The vertex v_{k+1} has at most one neighbor among $\{v_1, v_2, \dots, v_k\}$. If it does, then let v_i be such neighbor, and otherwise let v_i be arbitrary. Let $u \in V(G) - S$ be a neighbor of the unique vertex in $X_i \cap \partial S$, and let $S' := S \cup \{u\}$. If $k = n-1$, then the sets $X_1, X_2, \dots, X_k, \{u\}$ show that G has an F minor, in which case the theorem holds. Thus we may assume that $k < n-1$. Then $S' \in \mathcal{S}$ by the second stoppage axiom. It follows from $(*)$ that S' is $\mathcal{S}$-free, and the sets $X_1, X_2, \dots, X_k, \{u\}$ show that S' also satisfies (ii) and (iii) above, contrary to the maximality of S. □

Exercise 2.2.16. Let $n \geq 0$ be an integer, let F be a forest with vertex-set $\{v_1, v_2, \dots, v_n\}$, let G be a graph, and let $\mathcal{S}$ be a stoppage in G of order $n+1$. Then there exists an $\mathcal{S}$-free set $S \in \mathcal{S}$ and disjoint connected sets $X_1, X_2, \dots, X_n \subseteq S$ such that

(i) $|X_i \cap \partial S| = 1$ for all $i = 1, 2, \dots, n$ and
(ii) if $1 \leq i < j \leq n$ and $v_i \sim v_j$ in F, then $X_i \sim X_j$ in G.

Theorem 2.2.17. *Let G be a graph, let $n \geq 0$ be an integer, and let $\mathcal{S}_n$ be as in* Definition 2.2.13. *Then the following conditions are equivalent:*

(1) *G has a proper stoppage of order $n+1$,*
(2) *$V(G) \notin \mathcal{S}_n$,*
(3) *G has path-width at least n.*

Proof. (3) ⇒ (2) is trivial, and (2) ⇒ (1) follows from Lemma 2.2.14. Finally, to prove (1) ⇒ (3) we will prove the contrapositive. Let $(W_1, W_2, \dots, W_\ell)$ be a path-decomposition of G of width $< n$, and let $\mathcal{S}$ be a stoppage in G of order $n+1$. We must show that $\mathcal{S}$ is not proper. To that end we claim that $W_1 \cup W_2 \cup \dots \cup W_i \in \mathcal{S}$ for all $i = 0, 1, \dots, \ell$. We prove the claim by induction. For $i = 0$ it follows from the first stoppage axiom, and so we may assume that $S := W_1 \cup W_2 \cup \dots \cup W_{i-1} \in \mathcal{S}$ for some $i \leq \ell$. From Exercise 2.1.3 we deduce that $\partial S \subseteq W_{i-1} \cap W_i$. It follows from the second stoppage axiom that $W_1 \cup W_2 \cup \dots \cup W_i = S \cup W_i \in \mathcal{S}$, proving the claim. The claim implies that $V(G) = W_1 \cup W_2 \cup \dots \cup W_\ell \in \mathcal{S}$, and so $\mathcal{S}$ is not proper, as desired. □

We now introduce a variation of the cops-and-robber game in which the robber is invisible. This version of the game is equivalent to clearing the graph of some plague that immediately spreads along cop-free paths. Here is the formal definition. We say that a *search* in G is a sequence $X = (X_1, X_2, \dots, X_m)$ of subsets of $V(G)$ such that $X_1 = \emptyset$ and for all $i = 1, 2, \dots, m-1$, either $X_{i+1} \subseteq X_i$ or $X_i \subseteq X_{i+1}$. Thus X_i is the set of vertices occupied by the searchers at time i. We say that X is a *search with at most n searchers* if $|X_i| \leq n$ for all $i = 1, 2, \dots, m$. Let $B_0 = V(G)$, and for $i = 1, 2, \dots, m$ let B_i be the set of all vertices $v \in V(G)$ such that $G \backslash X_i$ has a path with one end v and the other end in B_{i-1}. Let $A_i = V(G) - B_i$. We say that the search X is *successful* if $B_m = \emptyset$, and if there is a successful search with at most n searchers then we say that *G can be searched by n searchers.* We say that the search X is search!cop-monotone if $X_i \cap X_k \subseteq X_j$ for all integers i, j, k with

$1 \leq i \leq j \leq k \leq m$, and we say that X is *robber-monotone* if $B_1 \supseteq B_2 \supseteq \cdots \supseteq B_m$. We say that the search is *monotone* if it is both cop-monotone and robber-monotone.

Exercise 2.2.18. If a search is cop-monotone, then it is robber-monotone.

Exercise 2.2.19. Let X be a successful search with at most n searchers, and let A_i be as above. Then
(i) $A_1 = \emptyset$, $A_m = V(G)$, and
(ii) $|(A_{i+1} - A_i) \cup \partial A_i| \leq n$ for all $i = 1, 2, \ldots, m-1$.

Definition 2.2.20. Let G be a graph, and let $n \geq 0$ be an integer. A *cleaning of G of breadth at most n* is a sequence $(A_1, A_2, \ldots, A_m)$ of subsets of $V(G)$ satisfying the two conditions of the previous exercise. The cleaning is *monotone* if $A_1 \subseteq A_2 \subseteq \cdots \subseteq A_m$.

Exercise 2.2.21. Let G be a graph. Then the function $X \mapsto |\partial X|$ is submodular; that is

$$|\partial A| + |\partial B| \geq |\partial(A \cap B)| + |\partial(A \cup B)|$$

for every $A, B \subseteq V(G)$.

Lemma 2.2.22. *Let G be a graph, and let $n \geq 0$ be an integer. If G has a cleaning of breadth at most n, then it has monotone cleaning of breadth at most n.*

Proof. Let $(A_1, A_2, \ldots, A_m)$ be a cleaning of breadth at most n chosen so that
(a) $\sum_{i=1}^{m} |\partial A_i|$ is minimum, and
(b) subject to (a), $\sum_{i=1}^{m} |A_i|$ is minimum.

We will show that $(A_1, A_2, \ldots, A_m)$ is monotone. To that end suppose for a contradiction that $A_i \not\subseteq A_{i+1}$ for some $i \in \{1, 2, \ldots, m\}$. Clearly $1 < i < m$. By Exercise 2.2.21 applied to $A = A_i$ and $B = A_{i+1}$ we deduce that either $|\partial(A_i \cap A_{i+1})| \leq |\partial(A_i)|$ or $|\partial(A_i \cup A_{i+1})| < |\partial(A_{i+1})|$.

Suppose first that $|\partial(A_i \cap A_{i+1})| \leq |\partial(A_i)|$, and let $A'_i := A_i \cap A_{i+1}$. Then $(A_1, A_2, \ldots, A_{i-1}, A'_i, A_{i+1}, \ldots, A_m)$ is a cleaning of G, because

$$|A_{i+1} - A'_i| = |A_{i+1} - A_i| \leq n - |\partial A_i| \leq n - |\partial A'_i|$$

and $|A'_i - A_{i-1}| \leq |A_i - A_{i-1}| \leq n - |\partial A_{i-1}|$. But that contradicts (a) or (b).

It follows that $|\partial(A_i \cup A_{i+1})| < |\partial A_{i+1}|$. Let $A'_{i+1} = A_i \cup A_{i+1}$. Then $(A_1, A_2, \ldots, A_i, A'_{i+1}, A_{i+2}, \ldots, A_m)$ is a cleaning of G, because $|A'_{i+1} - A_i| = |A_{i+1} - A_i| \leq n - |\partial A_i|$ and if $i + 2 \leq m$, then

$$|A_{i+2} - A'_{i+1}| \leq |A_{i+2} - A_{i+1}| \leq n - |\partial A_{i+1}| \leq n - |\partial A'_{i+1}|,$$

contradicting (a). □

Exercise 2.2.23. Let G be a graph, let $(A_1, A_2, \ldots, A_m)$ be a monotone cleaning of G, and let $W = (W_1, W_2, \ldots, W_{m-1})$ be defined by $W_i = (A_{i+1} - A_i) \cup \partial A_i$. Then W is a path-decomposition of G.

Lemma 2.2.24. *Let G be a graph, and let $n \geq 0$ be an integer. Then the conditions* (1)–(3) *of the previous lemma are equivalent to*
(4) *G cannot be searched with n searchers,*
(5) *G cannot be monotonely searched with n searchers.*

Proof. Clearly (4) $\Rightarrow$ (5). To prove (5) $\Rightarrow$ (3) we will prove the contrapositive. Let $(W_1, W_2, \ldots, W_\ell)$ be a path-decomposition of G of width $< n$. Then

$$W_1, W_1 \cap W_2, W_2, W_2 \cap W_3, \ldots, W_{\ell-1}, W_{\ell-1} \cap W_\ell, W_\ell$$

is a successful monotone search with at most n searchers, as desired.

Finally we prove that (1) $\Rightarrow$ (3), again by proving the contrapositive. Let $(X_1, X_2, \ldots, X_m)$ be a successful search with at most n searchers. By Exercise 2.2.19 the graph G has a cleaning of breadth at most n. By Lemma 2.2.22 there is a monotone cleaning of breadth at most n, and hence by Exercise 2.2.23 the graph G has path-width at most $n-1$, as desired. □

Definition 2.2.25. Let G be a graph, let $n \geq 1$ be an integer, and let $\mathcal{B}$ be a set of separations of G, all of order $< n$, such that

(i) if (A, B) is a separation of G of order $< n$, then $\mathcal{B}$ includes one of (A, B), (B, A), and

(ii) if (A_1, B_1), $(A_2, B_2) \in \mathcal{B}$, then $G[A_1] \cup G[A_2] \neq G$.

In those circumstances we say that $\mathcal{B}$ is a *blockage* in G of order n.

The following is proved in [**3**]. We omit the proof, because we do not need the result.

Theorem 2.2.26. *A graph has a blockage of order $n+1$ if and only if it has path-width at least n.*

Definition 2.2.27. Let M be an $n \times m$ 0-1 matrix. We define the *track-number* of M to be the minimum integer $k \geq 0$ such that there exists a matrix M' obtained from M by permuting columns with the following property. In every row of M' we find the left-most one and the right-most one, and change all zeros in between to ones. We do this for every row, and the property we are describing is that after this operation every column contains at most k ones. The GATE MATRIX LAYOUT problem asks, given k and M, if the track-number of M is at most k. This is an important problem in VLSI design. The columns of M correspond to *gates*. If there is a row with ones in columns corresponding to gates g_1 and g_2 then g_1 and g_2 must be *connected*. The gates will be placed consecutively next to each other on a long thin rectangular board, and the connections will be realized by means of parallel *tracks*. Obviously, we want to minimize the number of tracks.

Exercise 2.2.28. Prove that for every integer $k \geq 0$ and for every 0-1 matrix M, there is a graph G such that M has track-number at most k if and only if G has path-width at most $k-1$. Moreover, the graph G can be constructed in linear time.

2.3. Tree-Width and Algorithms

Theorem 2.3.1 ([**2**]). *Computing tree-width is NP-hard.*

Theorem 2.3.2 ([**4**]). *For every fixed integer k there is a linear-time algorithm to decide if an input graph has tree-width at most k.*

Theorem 2.3.3 ([**14**]). *There exists a polynomial-time algorithm that, given a graph G and an integer k, either finds a tree-decomposition of G of width at most k^4, or (correctly) answers that G has tree-width at least k.*

Remark 2.3.4. For every fixed integer k there is a dynamic programming algorithm with running time $O(|V(G)|^{k+2})$ to test if the input graph has tree-width $\leq k$, and if so, to find a tree-decomposition of width $\leq k$ [**2**]. This algorithm is relatively easy to implement.

Remark 2.3.5. An important feature of tree-width is that many NP-hard problems can be solved in linear time if the input graphs come with a tree-decomposition of width at most k, for some fixed integer k. Here is an outline of the general method.

Exercise 2.3.6. Let k be a fixed integer, let G be a graph, and let $Z \subseteq V(G)$ with $|Z| \leq k+1$. Suppose we want to compute certain information $P(G, Z)$ about G and Z, and this information satisfies the following axioms:

(i) $P(G, Z)$ can be computed in constant time if $|V(G)| \leq k+1$,

(ii) if $Z' \subseteq Z$ then $P(G, Z')$ can be computed from the knowledge of $P(G, Z)$ in constant time, and

(iii) if (G_1, G_2) is a separation of G with $V(G_1) \cap V(G_2) \subseteq Z$, then $P(G, Z)$ can be computed from the knowledge of $P(G_1, Z \cap V(G_1))$ and $P(G_2, Z \cap V(G_2))$ in constant time.

Then, given a tree-decomposition of G of width at most k, $P(G, \emptyset)$ can be computed in linear time.

Exercise 2.3.7. For every fixed integer k, the independence number of G can be computed in linear time if a tree-decomposition of G of width at most k is given.

Hint. For $A \subseteq Z$, let α_A be the maximum cardinality of an independent set $I \subseteq V(G)$ with $I \cap Z = A$ (or zero if no such set exists). Let $P(G, Z) = (\alpha_A : A \subseteq Z)$.

Exercise 2.3.8. For every fixed integer k, the chromatic number of G can be computed in linear time if a tree-decomposition of G of width at most k is given.

Exercise 2.3.9. Let G be a graph, and let $Z \subseteq V(G)$. Let us say that a *template* in Z is a sequence $\tau = (s_1, s_2, \ldots, s_k, t_1, t_2, \ldots, t_k)$ of vertices of Z such that the vertices $s_1, s_2, \ldots, s_k$ are pairwise distinct, and so are the vertices $t_1, t_2, \ldots, t_k$. We say that a template τ is *feasible* in G if there exist internally disjoint paths $P_1, P_2, \ldots, P_k$ in G such that P_i has ends s_i and t_i and $P_1 \cup P_2 \cup \cdots \cup P_k$ is an induced subgraph of G. Now let $P(G, Z)$ consist of all pairs $(H, \mathcal{T})$, where H is a simple graph with $V(H) = Z$ and $\mathcal{T}$ is a set of templates in Z such that there exists a chordal supergraph J of G such that H is the underlying simple grah of $J[Z]$ and $\mathcal{T}$ is the set of all templates in Z that are feasible in J. Prove that $P(G, Z)$ satisfies conditions (i)–(iii) of Exercise 2.3.6.

Exercise 2.3.10. For all fixed integers k and t find a linear-time algorithm that given a graph G and a tree-decomposition (T, W) of G of width at most k will either correctly establish that G has tree-width at least t, or will return a tree-decomposition of G of width at most $t-1$.

Exercise 2.3.11. For every fixed integer k and every fixed graph H, given an input graph G and a tree-decomposition of G of width at most k, one can determine whether G has an H minor in linear time.

2.4. Branch-Width

Branch-width is a variant of tree-width that can be extended to matroids.

Definition 2.4.1 ([**12**])**.** A tree T is *ternary* if every vertex of T has valency three or one. The vertices of valency one are called the *leaves* of T. Let G be a graph. A *branch-decomposition* of G is a pair (T, τ), where T is a ternary tree, and τ is a bijection between the set of leaves of T and $E(G)$. Let $e \in E(T)$, and let T_1, T_2 be the two components of $T \backslash e$. For $i = 1, 2$, let E_i be the set of all $\tau(t)$ for all leaves t of T with $t \in V(T_i)$. Then $E(G) = E_1 \cup E_2$, and we define the *order* of e to be the number of vertices of G incident with both E_1 and E_2. We define the *width* of (T, τ) to be the maximum order of the edges of T (or zero if $|E(G)| \leq 1$), and we define the *branch-width* of G to be the least integer $k \geq 0$ such that G has a branch-decomposition of width k, or zero if $|E(G)| \leq 1$, in which case G has no branch-decomposition.

Exercise 2.4.2. If G has an H-minor, then the branch-width of H is at most the branch-width of G.

Exercise 2.4.3. A graph G has branch-width zero if and only if every component of G has at most one edge.

Exercise 2.4.4. A graph G has branch-width at most one if and only if every component of G has at most one vertex of valency more than one.

Exercise 2.4.5. A graph G has branch-width at most two if and only if G is series-parallel.

Exercise 2.4.6 ([**12**])**.** For $n \geq 3$ the complete graph K_n has branch-width $\lceil \frac{2}{3} n \rceil$.

Exercise 2.4.7 ([**12**])**.** Let G be a graph, let t be the tree-width of G, and let b be the branch-width of G. If $b \geq 2$ then

$$b \leq t + 1 \leq \left\lfloor \frac{3}{2} b \right\rfloor .$$

Definition 2.4.8. Let G be a graph, and leg $k \geq 1$ be an integer. A *tangle* in G of order k is a set $\mathcal{T}$ of separations of G, each of order $< k$, such that

(i) for every separation (A, B) of G of order $< k$, one of (A, B), (B, A) is in $\mathcal{T}$,

(ii) if (A_1, B_1), (A_2, B_2), $(A_3, B_3) \in \mathcal{T}$, then $G[A_1] \cup G[A_2] \cup G[A_3] \neq G$, and

(iii) if $(A, B) \in \mathcal{T}$, then $A \neq V(G)$.

Theorem 2.4.9. *For every graph G of branch-width at least two the branch-width of G is equal to the maximum order of a tangle in G.*

Definition 2.4.10. Let M be a matroid. We define the *branch-decomposition* of M similarly as for graphs. We define the *order* of an edge $e \in E(T)$ to be $\kappa(E_1) = \kappa(E_2)$, where κ is the connectivity function of M and E_1, E_2 are as in Definition 2.4.1, and then we define the branch-width in the same way as for graphs.

Exercise 2.4.11. Let G be a graph. Prove that G has branch-width at most two if and only if $\mathcal{M}(G)$ has branch-width at most two. Notice that the two definitions use different definitions of an order of an edge.

Open Problem 2.4.12. Let G be a graph with branch-width at least two. Is it true that G and $\mathcal{M}(G)$ have the same branch-width?

2.5. Path-Width leftovers

Theorem 2.5.1 ([**3**]). *If F is a forest and G is a graph of path-width at least $|V(F)| - 1$, then G has an F minor.*

Theorem 2.5.2 ([**2**]). *Computing path-width is NP-hard.*

Definition 2.5.3. Let G be a graph. We define the *linear-width* of G to be the least integer $k \geq 0$ such that the edges of G can be numbered $e_1, e_2, \ldots, e_m$ in such a way that for every $i = 1, 2, \ldots, m-1$, there are at most k vertices incident with both $\{e_1, e_2, \ldots, e_i\}$ and $\{e_{i+1}, \ldots, e_m\}$. This again can be extended to matroids.

Exercise 2.5.4. What is the relation between path-width and linear-width?

Remark 2.5.5. Linear-width can be extended to matroids, but what should the analogue of Definition 2.5.1 be? The next result suggests a possible approach.

Theorem 2.5.6 ([**1**]). *Let Q be an outerplanar graph, and let R be a graph such that the deletion of some vertex of R produces a forest. For every such pair Q and R there exists an integer k such that every 2-connected graph of path-width at least k has a Q-minor or an R-minor.*

Definition 2.5.7 ([**3**]). Let $k \geq 0$ be an integer. A *stoppage* of order k is a set $\mathcal{S}$ of separations of order $< k$ such that

(i) if (A, B) is a separation of G of order $< k$, then $\mathcal{S}$ contains one of $(A, B), (B, A)$, and

(ii) if $(A_1, B_1), (A_2, B_2) \in \mathcal{S}$ then $A_1 \cup A_2 \neq G$.

If $(A, B) \in \mathcal{S}$ it helps to think of A as the "small" side and B as the "big" side of the separation.

Theorem 2.5.8 (Min-max theorem for path-width [**3**]). *Let $k \geq 0$ be an integer. A graph has a stoppage of order k if and only if it has path-width at least $k - 1$.*

Exercise 2.5.9. Let $k \geq 0$ be an integer, and let G be a graph. The following conditions are equivalent.

(i) G has a path-width $\geq k - 1$,

(ii) k cops can capture an invisible robber in G, and

(iii) G has a stoppage of order k.

Bibliography

[1] C. Anhalt and R. Thomas, Two-connected graphs of large path-width, manuscript.

[2] S. Arnborg, D. G. Corneil and A. Proskurowski, Complexity of finding embeddings in a k-tree, *SIAM J. Alg. Disc. Meth.* **8** (1987), 277-284.

[3] D. Bienstock, N. Robertson, P. D. Seymour and R. Thomas, Quickly excluding a forest, *J. Combin. Theory Ser. B* **52** (1991), 274–283.

[4] H. L. Bodlaender, A linear time algorithm for finding tree-decompositions of small tree-width, *J. Algorithms* **21** (1996), 358–402.

[5] R. Diestel, Graph Minors I: a short proof of the path-width theorem, *Combinatorics, Probability and Computing* **4** (1995), 27–30.

[6] R. Diestel, K. Yu. Gorbunov, T. R. Jensen, and C. Thomassen, Highly connected sets and the excluded grid theorem, *J. Combin. Theory Ser. B* **75** (1999), 61–73.

[7] M. Juvan, B.Mohar and R. Thomas, List edge-colorings of series-parallel graphs, *Electron. J. Combin.* **6** (1999), #R42.

[8] I. Kříž and R. Thomas, Clique-sums, tree-decompositions and compactness, *Discrete Math.* **81** (1990), 177–185.

[9] N. Robertson and P. D. Seymour, Graph Minors I. Excluding a forest, *J. Combin. Theory Ser. B* **35** (1983), 39–61.

[10] N. Robertson and P. D. Seymour, Graph Minors III. Planar tree-width, *J. Combin. Theory Ser. B* **36** (1984), 49–63.

[11] N. Robertson and P. D. Seymour, Graph Minors V. Excluding a planar graph, *J. Combin. Theory Ser. B* **41** (1986), 92–114.

[12] N. Robertson and P. D. Seymour, Graph Minors X. Obstructions to tree-decomposition, *J. Combin. Theory Ser. B* **52** (1991), 153–190.

[13] N. Robertson, P. D. Seymour and R. Thomas, Quickly excluding a planar graph, *J. Combin. Theory Ser. B* **62** (1994), 323–348.

[14] N. Robertson, P. D. Seymour and R. Thomas, unpublished.

[15] P. D. Seymour and R. Thomas, Graph searching, and a min-max theorem for tree-width, *J. Combin. Theory Ser. B* **58** (1993), 22–33.