Robertson-Seymour's minor theorem

Theorem 1 (Robertson-Seymour '04 [**?**])**.** *Graphs are well-quasi-ordered under the minor relation.*

This week we continue our work towards Robertson-Seymour and its implications for graph algorithms. We summarize the importance. We start with a definition.

Definition 2. *Let k be a parameter of the input of a problem, and n the size of the input. We say that a problem is* fixed parameter tractable (FPT) *if there exists an algorithm that solves it in time $f(k) \cdot n^{O(1)}$.*

There is no restriction on $f(k)$. In a sense, a problem that is FPT can capture its super-polynomial complexity in the parameter k rather than n. Then, a class of inputs that which has k bounded by a constant has a polynomial time algorithm even if n is unbounded.
A few notes on the significance of Robertson-Seymour for algorithmic complexity:

- Treewidth is the most significant example of a parameter for FPT problems. All of the graph problems that we have discussed (and all graph problems in Karp's original list of NP-complete problems) are FPT by treewidth: independent set, vertex cover, proper vertex coloring, Hamiltonicity,... These problems can be solved via dynamic programming on a tree decomposition of the graph. We will fill in one missing piece here: optimal tree decompositions can be found in polynomial time.
- Tree decompositions and treewidth are crucial tools in the proof of the Robertson-Seymour minor theorem. One step of their proof asserts that families of graphs of bounded treewidth are well-quasi-ordered by the minor relation, and another step characterizes the forbidden minors for the property of having bounded treewidth. In the study of graph algorithms, those minors are exactly the structures that are responsible for the complexity of the above-mentioned NP-complete problems.
- Another consequence of Robertson-Seymour is that any graph property that is inherited under taking minors can be decided in polynomial time. This follows from the fact that there is only a finite set of forbidden minors to check, together with the fact that minors can be checked in polynomial time. This implies immediately that none of the NP-complete in the list above are inherited under taking minors, but, for example, deciding whether a graph is planar is in this category.

Finding tree decompositions

The problem of determining the treewidth of a graph is NP-hard in general. However, if we set p a constant, then there exist algorithms that are polynomial in n that output a treedecomposition of width p if and only if such a decomposition exists. Here, we will prove a slightly weaker version of this statement, but this version is sufficient in that it outputs a tree decomposition of constant width if p is constant. The proof of Proposition 5 below is inspired by the proof given in Uriel Feige's lecture notes.

Definition 3. *For a graph G and (not necessarily disjoint) sets $A, B, S \subseteq V(G)$, we say that S is an A,B-*separator *in G if every path from a vertex in A to a vertex in B intersects S. Note that we must have $A \cap B \subseteq S$.*

Lemma 4. *Let t and p be constants. For G a graph with $\mathrm{tw}(G) = p$, and any set $W \subseteq V(G)$ with $|W| = t$, there exists a set S of cardinality at most $p+1$, such that $W \setminus S = W_1 \cup W_2 \cup \ldots W_t$, $1 \leq |W_i| \leq \frac{t}{2}$, with $W_i \cap W_j = \emptyset$ and S is a W_i, W_j-separator in G for $1 \leq i, j \leq t$. Furthermore, given W, such a set S can be found in polynomial time.*

Exercise 1. *Prove Lemma 4.*

Note that the proof above does not attempt to optimize the algorithm; it simply shows that a polynomial time algorithm exists.

Proposition 5. *If $\mathrm{tw}(G) \leq p$, then there exists a polynomial time algorithm that finds a tree decomposition of width $\leq 3(p+1)$.*

Proof. We will find a tree decomposition with bags of size at most $3(p+1)+1$. We work inductively from a partial tree decomposition T' of G with the following properties.

(i) T' is a valid tree decomposition of $G[V']$, the subgraph of G induced by the vertex set V', where
$$V' = \bigcup_{B \in V(T')} B,$$
and the bags of T' have cardinality at most $3(p+1)+1$.

(ii) For every connected component C of $G - V'$, there exists a bag $B \in V(T')$ that contains $N(C) \cap V'$, and $|N(C) \cap V'| \leq 2(p+1)+1$.

We will induct on $|V'|$. As the base case, one can let T' be a single bag containing any $2(p+1)+1$ vertices of G. Now, suppose that T' is any such partial tree decomposition of G, we will extend T' to a larger partial tree decomposition that contains at least one more vertex. Let C be a connected component of $G - V'$, let B be a bag containing its neighborhood, and let $X = N(C) \cap B$.

If $|X| \leq 2(p+1)$, let $B^* = X \cup \{v\}$ for any $v \in C$, and append B^* as a leaf to B. It is easy to check that this gives a larger, valid partial tree.

Suppose that $|X| = 2(p+1)+1$. As in Lemma 4, find a set S of cardinality at most $p+1$ such that $X \setminus S = X_1 \cup X_2 \cup \cdots \cup X_t$, $|X_i| \leq \lfloor \frac{2(p+1)+1}{2} \rfloor = p+1$ and S is a X_i, X_j-separator in G for $1 \leq i, j \leq t$. Note that $t \geq 2$, and since S is a X_1, X_2-separator, we have that $S \cap C \neq \emptyset$. Otherwise, there would be a X_1X_2-path via C in $G - S$. Let $S^* = S \cap (X \cup C)$, let $B^* = X \cup S^*$, and append B^* as a leaf to B. Note that the newly created connected components of $C \setminus S$ have their neighborhood contained in some $X_i \cup S^*$, and therefore at most $2(p+1)$ neighbors in B^*. Furthermore, $|B^*| = 3(p+1)+1$, since $|X| = |2(p+1)+1|$ and $|S| \leq p+1$. Therefore, we again see that we have a larger, valid partial tree. □

Detecting substructures

We have seen that the decision problem of whether a graph G contains a graph H as a subgraph is NP-hard in general. For example, consider the Hamiltonian cycle problem. We will show that the decision problem of whether a graph G contains a graph H as an induced subgraph/subgraph/minor is FPT in the order of H. This tells us that the decision problem

of whether G has a property that is closed under taking induced subgraphs/subgraphs/minors is in P if the property is finitely defined (by a list of finitely forbidden substructures). By Robertson-Seymour, this is always the case for minors. In all of these cases, we don't attempt to optimize the algorithms here, just to show that it can be done in polynomial time.

Let G be an n-vertex graph and H a k-vertex graph, where k is a constant. For induced subgraphs and subgraphs, $H \leq G$ is equivalent to the existence of a map $f : V(H) \to V(G)$, such that $vw \in E(H)$ implies $f(v)f(w) \in E(G)$, and in the case of induced subgraphs additionally that $vw \notin E(H)$ implies $f(v)f(w) \notin E(G)$. There are only n^k such maps, each of which is checked in constant time.

Exercise 2. *Let G be an n-vertex graph and H a k-vertex graph, where k is a constant. Give a polynomial time algorithm to check whether H is a minor of G.*

Exercise 3. *By the this method, we see that the independent/set and clique problems are in P if we parameterize by the size of the independent set. So, the decision problem of whether a graph G has an independent set (or a clique) of order k is in P when k is a constant. Show that the k-independent set (or clique) problem is also in P when $k = n - t$, and t is a constant.*

Exercise 4. *Then, let G be an n-vertex graph and H an $n - t$ vertex graph and consider the decision problem of whether H is a subgraph of G. Find a parameter of H by which this problem is FPT, or describe graph classes other than independent sets/cliques for which this problem is in P.*

Exercise 5. *Write a polynomial time algorithm for testing graph planarity. The best known algorithms run in $O(n)$ time (see Hopcroft-Tarjan), but focus on a simpler algorithm instead of optimizing for complexity.*