Tutorial 5: Treewidth

(1) Give three examples of graphs and their treewidths.

A tree decomposition T of width k is *smooth*, if for every $X_i \in T$, we have $|X_i| = k + 1$, and for every edge $(X_i, X_j) \in T$, we have $|X_i \cap X_j| = k$. Note that a smooth tree decomposition is also a nice tree decomposition.

(2) Prove that every tree decomposition can be transformed into a smooth one with the same width.

A graph H is a *minor* of G, if it is isomorphic to a graph that can be obtained from G by the following operations.

- Deleting an edge.
- Deleting an isolated vertex.
- Contracting an edge, i.e., removing an edge while simultaneously merging its two endpoints.

(3) Prove that if H is a minor of G, then $\mathrm{tw}(H) \leqslant \mathrm{tw}(G)$.

(4) Prove that $\mathrm{tw}(G) = 1$ if and only if K_3 is not a minor of G.

A graph is *series-parallel*, if it can be constructed inductively from a single vertex as follows.

- Adding a self-loop on a vertex.
- Adding a new edge parallel to an existing one.
- Adding a new vertex and a new edge connecting it to an existing vertex.
- Subdividing an existing edge (u, v), i.e., adding a new vertex w, and two new edges (u, v) and (v, w), while deleting the edge (u, v).

(5) Prove that $\mathrm{tw}(G) \leqslant 2$ if and only if G is a series-parallel graph.

Appendix

As far as I know, smooth decomposition is introduced in [1], which also contains a linear-time algorithm for constructing tree-decompositions of graphs of bounded treewidth.

References

[1] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996.