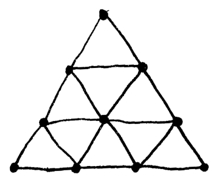
RWTH Aachen
Lehrgebiet Theoretische Informatik
Dreier—Kuinke—Rossmanith

WS 2018
Problem Set 4
12.11.2018

Parameterized Algorithms Tutorial

Tutorial Exercise T1

Determine the treewidth of the graph on the right.



Tutorial Exercise T2

Recall that a tree-decomposition of a graph $G = (V, E)$ is a pair $\langle T, \mathcal{X} = \{X_i \mid i \in V(T)\}\rangle$, where T is a tree whose vertices are called *nodes* and $\mathcal{X}$ is a collection of subsets of $V(G)$ called *bags* such that the following hold.

1. $\bigcup_{i\in V(T)} X_i = V(G)$.
2. For each edge $\{x, y\} \in E(G)$, there exists $i \in V(T)$ such that $x, y \in X_i$.
3. For all $i, j, k \in V(T)$, if X_j is in the path between X_i and X_k in the tree T, then $X_i \cap X_k \subseteq X_j$.

Show that the last condition can be replaced by the equivalent condition: For each vertex $u \in V(G)$, the set of bags that contain u is a subtree of T.

Tutorial Exercise T3

A tree-decomposition $\langle T, \mathcal{X} = \{X_i \mid i \in V(T)\}\rangle$ of a graph $G = (V, E)$ is *nice* if it is rooted at some node and has only four types of nodes.

1. *Leaf nodes* i, the leaves of the decomposition, with $|X_i| = 1$.
2. *Introduce nodes* i that have exactly one child j such that $X_i = X_j \cup x$, for some vertex $x \in V(G)$.
3. *Forget nodes* i that have exactly one child j such that $X_i = X_j \setminus x$, for some vertex $x \in V(G)$.
4. *Join nodes* i that have exactly two children j and k such that $X_i = X_j = X_k$.

Given a tree decomposition $\langle T', \mathcal{X}'\rangle$ of G of width w, construct a nice tree-decomposition $\langle T, \mathcal{X}\rangle$ of G in polynomial time of width w such that $|V(T)| = O(w \cdot |V(T')|)$.

Tutorial Exercise T4

The notion of treewidth can be defined in several ways. One way to frame the definition of treewidth is by using the following game called the *cops-and-robber* game. The game consists of a set of cops trying to catch a robber. The robber lives in the the graph and can move with infinite speed along the edges of the graph. He cannot, however, move through a vertex should a cop be guarding it. The cops move about in helicopters, the point being that they are not constrained to move along the edges of the graph, but they have finite speed. The game proceeds as follows. Initially, the robber occupies some vertex of the graph. The cops announce their positions (a set of vertices) and move towards them with finite speed. Seeing their positions, the robber announces his position (a vertex) and moves to that vertex instantaneously. Not all cops need land on vertices at once and not all cops need change positions, that is, if a cop occupies a vertex, it may continue occupying that vertex in the next move of the game. The cops *catch* the robber when one of them lands on a vertex occupied by him.

Show that if a graph has treewidth k then $k+1$ cops can always catch the robber in a cops-and-robber game on the graph. The converse also holds and this is not so easy to show. In the exercises, we will assume this equivalent formulation of treewidth.

Homework H1

Use the properties of a tree-decomposition to show that if a graph G contains a clique C, then every tree-decomposition $\langle T, \mathcal{X} \rangle$ of G has a bag X such that $V(C) \subseteq X$.

Homework H2

Let $\langle T, \mathcal{X} \rangle$ be a tree-decomposition of a graph G. Suppose that the subtrees obtained by deleting a node t from T are $T_1, \ldots, T_r$ and for $1 \leq i \leq r$, let G_i be the graph induced by the vertices of G that are in the bags of T_i. Then the subgraphs $G_1 - X_t, \ldots, G_r - X_t$ have no vertices in common and there are no edges between them.

Homework H3

Let $\langle T, \mathcal{X} \rangle$ be a tree-decomposition of a graph G. Suppose that the subtrees obtained by deleting an edge $\{i, j\} \in E(T)$ are T_i, T_j and let G_i, G_j be the subgraphs induced by the vertices in the bags of T_i and T_j, respectively. Show that deleting the set $X_i \cap X_j$ from $V(G)$ disconnects G into two subgraphs $G_i' := G_i - (X_i \cap X_j)$ and $G_j' := G_j - (X_i \cap X_j)$; that is, they do not share vertices and there is no edge with one end in each of them.