Jacob Focke and Philip Wellnitz Summer 2023

Techniques for Counting Problems, Exercise Sheet 4

www.mpi-inf.mpg.de/departments/algorithms-complexity/teaching/summer23/counting

Total Points: **100** Due: Sunday, **June 18**, 2023

You are allowed to collaborate on the exercise sheets. Justify your answers. Cite all external sources that you use (books, websites, research papers, etc.). You need to collect at least 50% of all points on exercise sheets to be admitted to the exam.
Please hand in your solutions before the lecture on the day of the deadline.

Exercise 1 — **25** points

For a graph G, we define a (nice) *tree decomposition* of G as a pair (T, β) of a rooted binary tree T and a mapping $\beta\colon V(T) \to 2^{V(G)}$ (a mapping from the nodes of T to subsets of $V(G)$ which we refer to as bags) such that

1. $\bigcup_{t \in V(T)} \beta(t) = V(G)$,
2. for every edge $uv \in E(G)$, there is some node $t \in V(T)$ such that $\{u, v\} \subseteq \beta(t)$,
3. for every vertex $v \in V(G)$, the set $\{t \in V(T) \mid v \in \beta(t)\}$ induces a connected subtree of T,
4. every node $t \in V(T)$ has exactly one of the following types:

   Leaf: t has no child and $\beta(t) = \emptyset$,

   Introduce: t has exactly one child t' and $\beta(t) = \beta(t') \cup \{v\}$ for some $v \in V(G)$; the vertex v is *introduced at* t,

   Forget: t has exactly one child t' and $\beta(t) = \beta(t') \setminus \{v\}$ for some vertex $V \in V(G)$; the vertex v is *forgotten at* t,

   Join: t has exactly two children t_1 and t_2 and $\beta(t) = \beta(t_1) = \beta(t_2)$, and
5. for the root node r of T we have $\beta(r) = \emptyset$.

We define the width of a tree decomposition as the size of its largest bag *minus one*. The treewidth of a graph is the minimum width of any of its tree decompositions.

Generalize the algorithm for counting homomorphisms from trees from the lecture to graph classes that contain only graphs of bounded treewidth. (A graph class $\mathcal{G}$ has bounded treewidth if there is a constant t such that every graph in $\mathcal{G}$ has a treewidth of at most t.)

Exercise 2 — **10 + 10** points

In the lecture, we started our reductions from a version of the clique problem that assumed that the host graph is connected. In this exercise, we show that this is not a restriction. In particular, prove the following (using different reductions), where in #Clique the host graph is not necessarily connected.

a #Clique $\leq_T^{fpt}$ #ConnectedClique.

b #Clique $\leq^{w\text{-}fpt}$ #ConnectedClique.

Exercise 3 — $10 + 10 + 10$ points

Recall the definition of a graph minor: a graph M is a minor of a graph H if M can be obtained from H via any combination of

deleting vertices of H, deleting edges of H, and contracting edges of H.

Prove that for any graphs M, H, and F, where M is a minor of H, there is a graph G with

$$\#\text{cp-Hom}(M \to F) = \#\text{cp-Hom}(H \to G),$$

and G can be constructed in time $O(|V(H)| \cdot |V(F)|)$.

Hint: Use induction and show that the above statement holds for each of the three possible modifications of the minor definition.

Exercise 4 — $10 + 15$ points

Prove that for any graph classes $\mathcal{H}$ and $\mathcal{G}$, we have

a $\#\textsc{Hom}(\mathcal{H} \to \mathcal{G}) \leq^{fpt} \#\textsc{cp-Hom}(\mathcal{H} \to \mathcal{G})$.

b $\#\textsc{cp-Hom}(\mathcal{H} \to \mathcal{G}) \leq^{fpt}_{T} \#\textsc{Hom}(\mathcal{H} \to \mathcal{G})$.

Hint: Recall that when counting color-prescribed homomorphisms, we are given graphs H and G and additionally a homomorphism $c \colon G \to H$ and we have to count homomorphisms $h \colon H \to G$ that satisfy $c(h(v)) = v$ for every vertex $v \in V(H)$. First observe that c has to be surjective and then count homomorphisms $g \colon H \to G$ so that $g \circ c$ is an automorphism of H (g is typically called "colorful homomorphism").
For the final step, use the Inclusion-Exclusion Principle.