Exercise sheet 11 – Solutions

COMP6741: Parameterized and Exact Computation

Serge Gaspers

Semester 2, 2017

Exercise 1. Show that Path Packing has no polynomial kernel unless NP $\subseteq$ coNP/poly.

Path Packing	
Input:	A graph G and an integer k
Parameter:	k
Question:	Are there k pairwise vertex-disjoint paths of length at least k each?

Solution. We give a polynomial parameter transformation from Long Path to Path Packing. Given an instance (G, k) to Long Path we construct a graph G' from G by adding $k-1$ vertex-disjoint paths of length k. Now, G contains a path of length k if and only if G' contains k vertex-disjoint paths of length k. The construction takes $O(k\cdot(n+m))$ time, where n and m are the number of vertices and edges of G, respectively, and the target parameter is k.

Exercise 2. An *endpoint* of a path is a vertex that has degree at most 1 in the path. Consider the NP-complete Anchored Path problem.

Anchored Path	
Input:	A graph $G = (V, E)$, a vertex $r \in V$, and an integer $k \leq \|V\|$
Parameter:	k
Question:	Does G have a path on k vertices as a subgraph such that r is an endpoint of that path?

Prove that Anchored Path has no polynomial kernel unless coNP $\subseteq$ NP/poly.

Solution. We give an OR-composition and use the Composition Theorem. To use the Composition Theorem, we need that Anchored Path is

- NP-complete (given in the question statement),
- the parameter can be computed in polynomial time (given in the input),
- and the value of the parameter is at most the instance size (guaranteed in the input specification).

Consider a finite sequence of instances $((G_i, r_i, k))_{1\leq i\leq t}$ for Anchored Path.
Construction. Our OR-composition algorithm computes an instance $(G, r, k+1)$ that is obtained by taking the disjoint union of all G_i, $1 \leq i \leq t$, adding a new vertex r that is adjacent to all r_i, $1 \leq i \leq t$.
Correctness. If some G_i has a path of length k starting from r_i, then adding r to the beginning of that path gives a path of length $k+1$ in G starting from r. On the other hand, if G has a path of length $k+1$ starting from r, the second vertex in that path is some r_i, and the following $k-1$ vertices of the path are all in G_i, and so G_i has a path of length k starting from r_i.
Running time. Clearly polynomial in the input size.