**RWTH Aachen** **WS 2018**
**Lehrgebiet Theoretische Informatik** Problem Set 3
Dreier—Kuinke—Rossmanith 5.11.2018

# Parameterized Algorithms Tutorial

## Tutorial Exercise T1

Given a graph $G = (V, E)$ and an integer $k$, you wish to add and/or delete a total of at most $k$ edges such that the modified graph consists of just one clique (and a set of isolated vertices). Design an FPT-algorithm for the problem with parameter $k$.

## Solution

First note that if a graph has an induced path on three vertices, then it cannot possibly be a disjoint union of a clique and a set of isolated vertices. In fact, a graph is a disjoint union of cliques if and only if it does not contain an induced $P_3$. Therefore if an instance to the above problem is a YES-instance, then one can first get rid of induced $P_3$s to obtain a disjoint union of cliques and then get rid of edges in all the other cliques save the largest one. This suggests a branching algorithm: we branch on induced $P_3$s and in a particular branch, we either choose to add the missing edge to the $P_3$ or remove one of the two edges. When there are no more induced $P_3$s, we delete edges from all cliques save the largest one. If we can do this within our budget of $k$ edge edit operations in any branch, we answer YES; else the instance is a NO-instance. The running time is $O(3^k \cdot \text{poly}(n))$.

## Tutorial Exercise T2

Give a polynomial kernel for the following problem.

| | |
|---|---|
| *Input:* | A sequence of marbles, each with a non-negative integer weight and color. |
| *Parameter:* | A positive integer $k$. |
| *Question:* | Can we remove marbles of total weight at most $k$, such that for each color, all marbles of that color are consecutive? |

## Solution

We use the following reduction rules: *Rule 1.* If there are two consecutive marbles of the same color, replace them by a single marble whose weight is the sum of the weights of the two marbles. Call a color *good* if it occurs only once. A marble is good if its color is good. *Rule 2.* If there are two consecutive marbles that are good, replace them with a single marble (with weight that is sum of the weights of the good marbles) and with color of, say, the first of them. Apply both these rules exhaustively. No two consecutive marbles have the same color and no two consecutive marbles are good. Observe that removing a marble can reduce the number of bad colors by at most two: the removed marble itself and the two marbles immediately to the left and right of the removed marble. This gives us *Rule 3:* If there are $2k + 1$ bad marbles, say NO. Finally, if a marble has weight at least $k + 2$, replace its weight by $k + 1$. The good marbles are between bad marbles and hence there are at most $2k + 2$ good marbles. Thus the resulting instance has a size of at most $(4k + 3)(k + 1) = O(k^2)$.

**Homework H1**

The parameterized DOMINATING SET problem is this: Given a graph $G = (V, E)$ and an integer $k$, decide whether there exists a vertex subset $S \subseteq V$ of size at most $k$ such that every vertex in $V \setminus S$ has a neighbor in $S$. Design an FPT-algorithm for this problem on graphs with maximum degree $d$, a constant.

**Solution**

The algorithm chooses a vertex $v$ that is not yet dominated and branches on its closed neighborhood (which is of size $d + 1$). In each branch, one vertex from the closed neighborhood of $v$ is selected in the solution; this vertex is deleted from the graph but all its neighbors are labeled "dominated." and the algorithm recurses on the resulting graph. The running time is $O((d + 1)^k \cdot \text{poly}(n))$.

**Homework H2**

You are given a boolean formula $\varphi$ in CNF such that maximum clause size is $q$, and an integer $k$. Design an FPT-algorithm with parameter $k$ that decides whether such a boolean formula has a satisfying assignment of weight at most $k$.

**Solution** We start with the assignment which sets all variables to FALSE. It has weight zero. We pick a clause wich is not satisfied. It contains only positive literals. We branch over all literals in this clause. In each branch one variable is set to TRUE. We can remove all clauses in which this variable occurs positively, as they are satisfied. We remove the variable from all clauses where it occurs negatively. If a clause is empty, we can reject this branch. Otherwise we recurse on the new formula. The time taken is $O(q^k \cdot \text{poly}(n))$.

**Homework H3**

Show that the following problem is in FPT by the method of reducing to a problem kernel.

| | |
|---|---|
| *Input:* | A set $S = \{x_1, \ldots, x_n\}$ and a collection $\mathcal{C}$ of subsets of $S$. |
| *Parameter:* | A positive integer $k$. |
| *Question:* | Is there a collection $\mathcal{B}$ of subsets of $S$ with $\lvert\mathcal{B}\rvert \leq k$ such that for each $A \in \mathcal{C}$, there exists a subcollection of $\mathcal{B}$ whose union is exactly $A$? |

(**Hint.** In a yes-instance, each set $A_i$ can be expressed a union of sets in $\mathcal{B}$. How many sets $A_i$ can there be? Also if two distinct elements $x_i$ and $x_j$ occur in exactly the same sets of $\mathcal{C}$, how may we handle them?)

**Solution**

Since in a yes-instance each $A \in \mathcal{C}$ can be represented uniquely by the union of sets from $\mathcal{B}$, $\mathcal{C}$ can have at most $2^k$ members. Moreover, if two distinct elements $x_i$ and $x_j$ occur in the same sets of $\mathcal{C}$, one of them can be removed. So how many elements can $S$ have? Consider a bipartite graph, the left partite set of which represents the set $S$ and the right partite set represents the set $\mathcal{C}$. A vertex $x$ in the left set has an edge to a vertex $A$ in the right set iff $x \in A$. We know that the size of the right set is at most $2^k$; that there are no isolated vertices in the left set; and no two vertices in the left set have the same neighborhood in the right set. The size of the left partite set is at most $2^{2^k}$.