Parameterized Algorithms WS 2021
Prof. Dr. P. Rossmanith
Dr. E. Burjons, M. Gehnen, H. Lotze, D. Mock

Date: November 29th, 2021

Exercise Sheet with solutions 06

Task T18

Show that Dominating Set $\leq_{\text{FPT}}$ Hitting Set.

Solution

Let (G, k) be an instance of Dominating Set. Create an instance $(\mathcal{U}, \mathcal{F}, k')$ of Hitting Set as follows. Set $\mathcal{U} = V(G)$ and $\mathcal{F} = \{N[u] \mid u \in V(G)\}$, and $k' = k$. One can easily verify that G has a k-dominating set iff $(\mathcal{U}, \mathcal{F})$ has a k-hitting set.

Task T19

Show that Clique $\leq_{\text{FPT}}$ Independent Set on regular graphs.

Solution

Given an instance (G, k) for Clique, we construct an instance (G', k) with a regular graph G' as dercribed below. If $k \leq 2$, then the Clique problem is trivial, hence we can output a trivial yes- or no-instance. Let d be the maximum degree of G.

1. Take d distinct copies $G_1, \ldots, G_d$ of G and let v_i be the copy of $v \in V(G)$ in graph G_i.
2. For every vertex $v \in V(G)$, let us introduce a set V_v of $d - d_{G(v)}$ vertices and add edges between every vertex of V_v and every v_i for $1 \leq i \leq d$.

Observe that every vertex of G' has degree exactly d. To prove the correctness of the reduction, we claim that G has a k-clique if and only if G' has. The left to right implication is clear: copies of G appear as subgraphs in G', thus any clique in G gives a corresponding clique in G'. For the reverse direction, observe that the vertices introduced in step 2. do not appear in any triangles. Therefore, assuming $k \geq 3$, these vertices cannot be part of a k-clique. Removing these vertices gives d disjoint copies of G, thus any k-clique appearing there implies the existence of a k-clique in G.
Finally take $(\bar{G}', k)$ where $\bar{G}'$ is the complementary graph to G', as our independent set instance. Because the regularity of the graph is not specified in the instance this is a valid fpt reduction from clique.

Task T20

Is there a parameterized reduction from Vertex Cover to Independent Set?

Solution

The following algorithm satisfies the formal definition of a parameterized reduction: solve the Vertex Cover instance in FPT time and output a trivial yes-instance or no-instance of Independent Set. More generally, if a problem A is FPT, then A has a parameterized reduction to any parameterized problem B that is nondegenerate in the sense that it has at least one yes-instance and at least one no-instance.

Task T21

Provide an FPT-reduction from Independent Set to Short Turing Machine Acceptance (STMA).

Solution

In this solution and the ones that follow, we will not be very formal in the Turing machine constructions. The goal is to get a feeling as to why such a reduction should work and not get bogged down in the details of Turing machine constructions.

Let (G, k) be an instance of the Independent Set problem. Construct a non-deterministic Turing machine T_G whose input alphabet consists of $n+1$ symbols $\{1, \ldots, n, \#\}$, where $n = |V(G)|$, and whose tape alphabet consists of the blank symbol $\{B\}$ and which works as follows:

1. The machine writes k symbols on its tape from the set $\{1, \ldots, n\}$.
2. It then verifies that the symbols written are distinct.
3. It then constructs the subgraph G' of G induced by these k vertices.
4. Finally, it verfies whether G' has edges and if not, it accepts.

Steps 1 and 2 take time $O(k)$ and $O(k^2)$. Assuming that the graph G is "hardwired" in the machine as an adjacency matrix, Steps 3 and 4 together take time $O(k^2)$. The size of the state space of the machine M and the transition function table can be seen to be polynomial in the size of G. A very rough estimate is as follows: $O(kn)$ states for choosing k vertices; $O(k)$ states for verifying whether the vertices chosen are all distinct; $O(k^2)$ states for constructing the subgraph and another $O(k^2)$ states for verifying whether the subgraph has any edges.

Task H13 (5 credits)

Show that Hitting Set $\leq_{\text{FPT}}$ Dominating Set.

Solution

Given an instance $(\mathcal{U}, \mathcal{F}, k)$ of Hitting Set, construct a graph $G = (V, E)$ as follows. Define $V(G) = \{x, y_1, \ldots, y_{k+1}\} \cup U \cup F$, where $u_i \in U$ for each element $i \in \mathcal{U}$ and $s_j \in F$ for each set $S_j \in \mathcal{F}$, and $x, y_1, \ldots, y_{k+1}$ are special vertices. Vertex $u_i \in U$ is connected to $s_j \in F$ iff $i \in S_j$ and vertex x is connected to every vertex $u_i \in U$ and to the vertices $y_1, \ldots, y_{k+1}$. The graph G has nor more edges.

We claim that there exists a hitting set of size k iff G has a dominating set of size $k+1$. Suppose that there exists a hitting set of size k. Choose the "corresponding" vertices from U and, in addition, choose vertex x. These $k+1$ vertices clearly dominate all vertices of G. If G has a dominating set of size $k+1$, then this set must include x. For otherwise, one would have to choose $y_1, \ldots, y_{k+1}$ to dominate all of these. Since x also dominates all vertices in U, clearly this set does not contain any vertex from F. This is because vertices in F can dominate vertices of U only. Hence there exists k vertices in U that dominate all of F. The "corresponding" elements of $\mathcal{U}$ hit all sets in $\mathcal{F}$ and hence there exists a hitting set of size k.

Task H14 (5 credits)

Provide an FPT-reduction from Dominating Set to Short Multi-Tape Turing Machine Acceptance.

Solution

Given an instance (G, k) of Dominating Set, construct a machine with $n + 1$ tapes, where $n = |V(G)|$. The input alphabet of the machine is $\{1, \ldots, n, \#\}$ and the tape alphabet is $\{B\}$, denoting the blank symbol. The machine works as follows:

1. It first writes down symbol i (denoting the ith vertex) on the ith tape.
2. Over a set of k moves, it chooses k vertices non-deterministically and writes these on to the $n + 1$st tape.
3. It verifies that the k vertices chosen are distinct.
4. It then moves the head of the $n + 1$st tape to the starting position (where it started writing down the candidate vertices).
5. Suppose that the tape-head on the $(n + 1)$st tape sees the jth vertex on that tape. If vertex i on tape i is dominated by the jth vertex on tape $n + 1$, and the ith tape head does not see a $\#$, the machine moves the ith tape head one cell to the right and prints a $\#$ on that cell. For a fixed j on tape $n + 1$, the machine does this simultaenously for all i satisfying this condition. Finally, the machine moves the tape-head on the $(n + 1)$st tape one cell to the right.
6. The machine accepts iff all tape heads from 1 to n see a $\#$.

The total time taken by the machine is $O(k^2)$. One can again verify that the number of states and the size of the transition function table is polynomial in n.