RWTH Aachen | **SS 2013**

Lehrgebiet Theoretische Informatik | Problem Set 7

Reidl—Rossmanith—Sánchez—Sikdar | 02.07.2013

Parameterized Algorithms Tutorial

In this tutorial we will revisit the basic technique of proving that certain problems that are fixed-parameter tractable do not admit polynomial kernels. In this context, recall that a *composition algorithm* for a parameterized problem $L \subseteq \Sigma^* \times \mathbf{N}$ is an algorithm that

- receives as input a sequence $((x_1, k), \dots, (x_t, k))$, with $(x_i, k) \in \Sigma^* \times \mathbf{N}$ for each $1 \le i \le t$;
- uses time polynomial in $\sum_{i=1}^{t} |x_i| + k$;
- outputs $(y, k') \in \Sigma^* \times \mathbf{N}$ such that k' is polynomial in k and $(y, k') \in L$ iff $(x_i, k) \in L$ for some $1 \le i \le t$.

Then the basic result that we use to prove the non-existence of polynomial kernels is: *A compositional parameterized problem whose unparameterized version is NP-complete does not admit a polynomial kernel (modulo some complexity-theoretic hypothesis).*

Tutorial Exercise T17

Suppose that L is a parameterized graph problem such that for any pair of graphs G_1 and G_2 we have that $(G_1, k) \in L \vee (G_2, k) \in L$ iff $(G_1 \uplus G_2, k) \in L$, where $G_1 \uplus G_2$ is the disjoint union of G_1 and G_2. Show that L is compositional. Conclude that the problems k-Path, k-Cycle, k-Exact Cycle and k-Short Cheap Tour are compositional.

Proposed Solution

Let $(G_1, k), \dots, (G_t, k)$ be instances of L. Create a composite instance $\widetilde{G}$ by taking the disjoint union $G_1 \uplus G_2 \uplus \dots G_t$ of all input instances and keeping the parameter as k. Then

$$\begin{aligned}
(G_1 \uplus \cdots \uplus G_t, k) \in L &\iff (G_1 \uplus \cdots \uplus G_{t-1}, k) \in L \vee (G_t, k) \in L \\
&\iff (G_1 \uplus \cdots \uplus G_{t-2}, k) \in L \vee (G_{t-1}, k) \in L \vee (G_t, k) \in L \\
&\ \ \vdots \\
&\iff (G_1, k) \in L \vee \cdots \vee (G_t, k) \in L.
\end{aligned}$$

Note that k-Path, k-Cycle, k-Exact Cycle and k-Short Cheap Tour all satisfy this property and hence are compositional.

Tutorial Exercise T18

Show that the Minor Order Test problem does not have a polynomial kernel (it is in FPT). An input to this problem consists of graphs $G = (V, E)$ and $H = (V', E')$, the parameter being the graph H. The question is to decide whether $H \preceq_{\text{minor}} G$.

Proposed Solution

If H is a connected graph then Minor Order Test is compositional by the previous exercise. We now reduce from the connected case to the disconnected case. Let (G, H) be an instance where H is connected. To create an instance where the minor being sought after is disconnected, consider the instance $(G \uplus G, H \uplus H)$. It is clear that $H \preceq_{\text{minor}} G$ iff $H \uplus H \preceq_{\text{minor}} G \uplus G$. By Exercise T20, the disconnected version does not have a polynomial kernel too.

Tutorial Exercise T19

Consider the **tw**-Independent Set problem: given a graph G, a tree-decomposition $\mathcal{T}$ of G of width w and an integer k, decide whether G has an independent set of size at least k. This problem is fixed-parameter tractable wrt w as parameter. Our goal is to show that this problem does not admit a polynomial kernel. To do this, first show that the problem **tw**-Independent Set Refinement, defined as follows, does not admit a polynomial kernel. Given a graph G, a tree-decomposition $\mathcal{T}$ of G of width w, and an independent set I of G, decide whether G has an independent set of size at least $|I| + 1$, where w is the parameter. Use this result to show that **tw**-Independent Set does not admit a polynomial kernel.

Proposed Solution

Let us first see that **tw**-Independent Set Refinement is compositional. Given t instances $((G_1, I_1, \mathcal{T}_1, w), \ldots, (G_t, I_t, \mathcal{T}_t, w))$, we simply take the disjoint union of these to create our composite instance $(\widetilde{G}, \widetilde{I}, \widetilde{\mathcal{T}}, w)$

$$\begin{aligned}\widetilde{G} &= G_1 \uplus \cdots \uplus G_t \\ \widetilde{I} &= I_1 \uplus \cdots \uplus I_t \\ \widetilde{T} &= \text{ connect all } \mathcal{T}_i \text{ arbitrarily to a tree}\end{aligned}$$

Note that this does not increase the treewidth! If $(\widetilde{G}, \widetilde{I}, \widetilde{\mathcal{T}}, w)$ has a solution of size $|\widetilde{I}| + 1$ then one of the graphs G_i has an independent set larger than I_i. The other direction is easy. We conclude that **tw**-Independendet Set Refinement does not admit a polynomial kernel (modulo strange complexity-theoretic events). It remains to show that **tw**-Independent Set does not admit a polynomial kernel. We proceed as follows: assume that **tw**-Independent Set has a polynomial kernel. We prove a parameter-preserving reduction from **tw**-Independent Set Refinement to **tw**-Independent Set which will prove that the former has a polynomial kernel, a contradiction!

Assume we are given an instance $(G, I, \mathcal{T}, w)$ of **tw**-Independent Set Refinement. We simply discard I to create an instance $(G, \mathcal{T}, w)$ of **tw**-Independent Set. The assumed kernelization algorithm reduces this to $(G', \mathcal{T}', w')$ with $|G'|, |\mathcal{T}'|, w' \leq \mathit{poly}(w)$. The trick now is that, because both problems are NP-complete, that we *know* that there exists a polynomial-time reduction between their unparameterized versions. Simply "forgetting" the parameterization (w is now simply part of the input) maps this instance back to $(G'', I'', \mathcal{T}'', w'')$. As this reduction happens in polynomial-time wrt to the input, we conclude that

$$|G''|, |I''|, |\mathcal{T}''|, |w''| \leq \mathit{poly}(|G'| + |\mathcal{T}'| + |w|) \leq \mathit{poly}(w),$$

which means that this instance is indeed a kernel of polynomial size. Here, it is crucial that the parameter in a parameterized instance is provided in *unary*, otherwise such a polynomial-time reduction could blow up the parameter exponentially.

This shows that **tw**-INDEPENDENT SET does not admit a polynomial kernel.

Tutorial Exercise T20

We generalize an observation from the last exercise. Suppose that A and B are parameterized problems such that the unparameterized version of A is NP-complete and that of B is in NP. Also suppose that there is a polynomial-time algorithm $\mathcal{A}$ that takes an instance (x, k) of A and, in time polynomial in $|x| + k$, outputs an instance (y, k') of B such that k' is polynomial in k and $(x, k) \in A$ iff $(y, k') \in B$. Show that if B admits a polynomial kernel then so does A. What can we say if A is NP-complete and compositional? The algorithm $\mathcal{A}$ is called a polynomial-time parameter-preserving reduction.

Proposed Solution

We can proceed as in T24: given (x, k), we obtain (y, k') and employ the kernelization to get (y', k'') with $|y'|, k'' \leq \mathit{poly}(k') \leq \mathit{poly}(k)$. Via a polynomial-time reduction we reduce this problem back to an instance (x', k''') of A with $|x'|, k''' \leq \mathit{poly}(k)$ which constitutes a kernel for A.

If now A is NP-complete and compositional, i.e. we do not expect A to have a polynomial kernel, then we can say the same about B: otherwise we could first use the parameter-preserving reduction from A to B, kernelize and then reduce back to A. This would provide a kernel for A, contradicting our assumption.

Homework H13

Show that the following problems are compositional.

1. w-CLIQUE: An input consists of a graph G, a tree-decomposition of G of width w, and an integer k. The problem is to decide whether G contains a clique of size k. The parameter is the width w.

2. k-LEAF OUT-TREE: An out-tree is a directed graph with a distinguished root vertex such that the underlying graph is a tree and all arcs are oriented away from the root. This problem is defined as follows: given a directed graph D and an integer parameter k, does D have as a subdigraph an out-tree with at least k leaves.

[10 points]

Homework H14

Show that the problem w-DOMINATING SET does not admit a polynomial kernel. This problem is defined similar to w-INDEPENDENT SET: given a graph G, a tree-decomposition of G of width w, an integer k, decide whether G has a dominating set of size k. The parameter is the width w. [Hint: Use the same strategy as for w-INDEPENDENT SET.]

[10 points]

Homework H15

Show that the following problem does not admit a polynomial kernel. k-Disjoint Paths: given a graph G and an integer parameter k, decide whether G has k vertex disjoint paths, each of length k. [Hint: Give a polynomial-time parameter-preserving reduction from k-Path.]

[10 points]