RWTH Aachen **WS 2011/12**
Lehrgebiet Theoretische Informatik Problem Set 9
Langer—Reidl—Rossmanith—Sikdar 13.01.2012

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 *distillation algorithm* for a decision problem $L \subseteq \Sigma^*$ is an algorithm that

- recieves as input a sequence $(x_1, \ldots, x_t)$, with $x_i \in \Sigma^*$ for each $1 \le i \le t$;
- uses time polynomial in $\sum_{i=1}^{t} |x_i|$;
- outputs a string $y \in \Sigma^*$ of length polynomial in $\max_{1 \le i \le t} |x_i|$ such that $y \in L$ iff $x_i \in L$ for some $1 \le i \le t$.

It is known that it is implausible for NP-complete problems to admit distillation algorithms (the exact complexity-theoretic collapse is not that relevant for us).
Also recall that a *composition algorithm* for a parameterized problem $L \subseteq \Sigma^* \times \mathbf{N}$ is an algorithm that

- receives as input a sequence $((x_1, k), \ldots, (x_t, k))$, 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 T23

Show that the problems k-Path and k-Cycle are compositional.

Proposed Solution

Given instances $((G_1, k), (G_2, k), \ldots, (G_t, k))$ we create a composite instance $\tilde{G}$ by taking the disjoint union of all input instances $\tilde{G} = G_1 \uplus G_2 \uplus \ldots G_t$ and keepin the parameter k.
It is easy to see that this instance contains a k-path or -cycle iff one of the graphs $G_1, \ldots, G_t$ contains one.

Tutorial Exercise T24

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.

Note: We renamed w-Independent Set to **tw**-Independent Set to avoid confusion

Proposed Solution

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

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

Note that this does not increase the treewidth! Now, in this instance, if we can find a solution set that is larger than the provided $\tilde{I}$, clearly that means that one of the graphs G_i has a larger independent set than the provided I_i. The other direction is even simpler.

We conclude that **tw**-Independet Set Refinement does not admit a polynomial kernel (modulo strange complexity-theoretic events). It remains to show that therefore **tw**-Independent Set does not admit a polynomial kernel. We will proceed as follows: assume there is a polynomial kernel. Then, through some reductions, this implies that **tw**-Independet Set Refinement would have a poly-kernel, contradicting the above.

Assume we are given an instance $(G, I, \mathcal{T}, w)$ of **tw**-Independet 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 $|G'|, |\mathcal{T}'|, w' \leq poly(w)$. 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

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

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 paramter by exponentially.

This therefore showed that **tw**-Independet Set does not admit a polynomial kernel.

Tutorial Exercise T25

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 B 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 $|y'|, k'' \leq poly(k') \leq poly(k)$. Via a polynomial-time reduction we reduce this problem back to an instance (x', k''') of A with $|x'|, k''' \leq poly(k)$ which constitutes a kernel.

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

Homework H19

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.

Proposed Solution

Both problems are compositional by simply taking the disjoint union of all instances.

Homework H20

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.]

Proposed Solution

The same as for T24.

Homework H21

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.]

Proposed Solution

We can easily reduce an instance of k-Path to one of k-Disjoint Paths as follows: given (G, k), create (G', k) by adding $k-1$ disjoint paths of length k to G. It is easy to see that G' contains k vertex-disjoint paths of length k iff G contains at least one path of length k.

As argued in the solution of T24, such a reduction means that the non-existence of polynomial kernels for k-Path carries over to k-Disjoint Paths.