

Exercise sheet 7 – Solutions and Hints

COMP6741: Parameterized and Exact Computation

Serge Gaspers

19T3

Exercise 1. A *domatic k-partition* of a graph $G = (V, E)$ is a partition (D_1, \dots, D_k) of V into k dominating sets of G .

(sol+tw)-DOMATIC PARTITION

Input: graph G , integer k

Parameter: $k + \text{tw}(G)$

Question: Does G have a domatic k -partition.

- Show that (sol+tw)-DOMATIC PARTITION is FPT using Courcelle’s theorem

Solution. To show that (sol+tw)-DOMATIC PARTITION is FPT, we express it as an MSO sentence which is true for the input graph G if and only if G has a domatic k -partition:

$$\exists D_1 \subseteq V. \exists D_2 \subseteq V. \dots \exists D_k \subseteq V. \text{partition}(D_1, D_2, \dots, D_k) \wedge \forall v \in V \text{dom}(v, D_1) \wedge \dots \wedge \text{dom}(v, D_k)$$

with

$$\begin{aligned} \text{partition}(D_1, \dots, D_k) := & \forall v \in V. (v \in D_1 \wedge v \notin D_2 \wedge v \notin D_3 \wedge \dots \wedge v \notin D_k) \vee \\ & (v \notin D_1 \wedge v \in D_2 \wedge v \notin D_3 \wedge \dots \wedge v \notin D_k) \vee \\ & \dots \\ & (v \notin D_1 \wedge v \notin D_2 \wedge v \notin D_3 \wedge \dots \wedge v \in D_k) \end{aligned}$$

and

$$\text{dom}(v, X) := v \in X \vee (\exists x \in X. \text{adj}(v, x))$$

The length of this expression is $O(k^2)$. Since this is a parameterized reduction to Courcelle’s problem, the result follows.

Exercise 2. Show that the incidence treewidth of a CNF formula F is at most the dual treewidth of F plus 1.

Solution. Start from a tree decomposition (T, γ) of the dual graph of F with minimum width. For each variable v in F , select a bag i_v that contains all the clauses where v occurs. Such a bag necessarily exists, since these clauses form a clique in the dual graph. Add a new bag containing v and all the clauses where v occurs, and make this bag adjacent to i_v . This gives a tree decomposition for the incidence graph of F whose width equals the width of the tree decomposition of the dual graph plus one.

Exercise 3. Show that CSP is W[1]-hard for parameter incidence treewidth and Boolean domain ($D = \{0, 1\}$).

Hints. Reduce from CLIQUE.

(1) Use Boolean variables x_{ij} with $1 \leq i \leq k$ and $1 \leq j \leq n$ with the meaning that x_{ij} is set to 1 if the i th vertex of the clique corresponds to the j th vertex in the graph.

(2) Add $O(k^2)$ constraints enforcing that for each $i \in \{1, \dots, k\}$, exactly one x_{ij} is set to 1, and whenever two $x_{ij}, x_{i'j'}$ with $i \neq i'$ are set to 1, then vertices j and j' are adjacent in the graph.

(3) Show that a graph with a vertex cover of size q has treewidth at most q .

Exercise 4. Design an $O^*(2^t)$ time DP algorithm for tw-INDEPENDENT SET.

tw-INDEPENDENT SET

Input: Graph G , integer k , and a tree decomposition of G of width t
 Parameter: t
 Question: Does G have an independent set of size k ?

Solution sketch.

- Obtain a nice tree decomposition (T, γ) of width t in polynomial time.
- Denote T_i the subtree of T rooted at node i
- Denote $\gamma_{\downarrow}(i) = \{v \in \gamma(j) : j \in V(T_i)\}$ and $G_{\downarrow}(i) = G[\gamma_{\downarrow}(i)]$
- For each node i of T , and each $S \subseteq \gamma(i)$, compute $\text{ind}(i, S)$, the size of a largest independent set of $G_{\downarrow}(i)$ that contains all vertices of S and no vertex from $\gamma(i) \setminus S$ by dynamic programming.
- For a leaf node i with $\gamma(i) = \{v\}$:

$$\begin{aligned}\text{ind}(i, \emptyset) &= 0 \\ \text{ind}(i, \{v\}) &= 1\end{aligned}$$

- For a forget node i with child i' and $\gamma(i) = \gamma(i') \setminus \{v\}$:

$$\text{ind}(i, S) = \max(\text{ind}(i', S), \text{ind}(i', S \cup \{v\}))$$

- For an introduce node i with child i' and $\gamma(i) = \gamma(i') \cup \{v\}$:

$$\text{ind}(i, S) = \begin{cases} -\infty & \text{if } G[S] \text{ contains an edge} \\ \text{ind}(i', S \setminus \{v\}) + [1 \text{ if } v \in S] & \text{otherwise} \end{cases}$$

- For a join node i with children i' and i'' :

$$\text{ind}(i, S) = \text{ind}(i', S) + \text{ind}(i'', S) - |S|$$

Exercise 5. Design an $O^*(9^t)$ time DP algorithm for **tw-DOMINATING SET**. Can you even achieve an $O^*(4^t)$ time DP algorithm?

tw-DOMINATING SET

Input: Graph G , integer k , and a tree decomposition of G of width at most t
 Parameter: t
 Question: Does G have a dominating set of size k ?

Solution sketch.

- Obtain a nice tree decomposition (T, γ) of width t in polynomial time.
- Denote T_i the subtree of T rooted at node i
- Denote $\gamma_{\downarrow}(i) = \{v \in \gamma(j) : j \in V(T_i)\}$
- Denote $G_{\downarrow}(i) = G[\gamma_{\downarrow}(i)]$
- For each node i of T , and each labelling $\ell : \gamma(i) \rightarrow \{\text{in}, \text{outD}, \text{outND}\}$, compute the smallest size of a subset D of $\gamma_{\downarrow}(i)$ such that $D \cap \gamma(i)$ is the set of vertices labelled **in** by ℓ , and that dominates all vertices from $\gamma_{\downarrow}(i)$ except those that are labeled **outND** by ℓ by dynamic programming.

The running time depends on how join nodes are handled. See Section 10.5 in the [Nie06] textbook for details.

References

[Nie06] Rolf Niedermeier. *Invitation to Fixed Parameter Algorithms*. Oxford University Press, 2006.