Lesson 7: Query evaluations on bounded treewidth graphs

Theme: Evaluating Boolean queries on bounded treewidth graphs

1 Hard problems that become easy for graphs with bounded treewidth

Clique. A clique in a graph G is a set of vertices in G that are adjacent to one another.

Input: A graph G of treewidth $\leqslant k$ and an integer M.

Task: Determine whether G has a clique of size $\geqslant M$.

Vertex cover. A vertex cover of a graph $G = (V, E)$ is a set $S \subseteq V$ such that every edge $(u, v) \in E$ is adjacent to at least one vertex in S. Consider the following problem $\textsc{Vertex-Cover}_k$.

Input: A graph G of treewidth $\leqslant k$ and an integer M.

Task: Determine whether G has a vertex cover of size $\leqslant M$.

Dominating set. A dominating set of a graph $G = (V, E)$ is a set $D \subseteq V$ such that every vertex in G is adjacent to at least one vertex in D. Consider the following problem $\textsc{Dominating-Set}_k$.

Input: A graph G of treewidth $\leqslant k$ and an integer M.

Task: Determine whether G has a dominating of size $\leqslant M$.

2 A meta theorem for evaluating MSO sentences on graphs

MSO sentences on graphs can be defined similarly as in Tutorial 6, i.e., by allowing quantification over sets of vertices: $\forall X \varphi$ and $\exists X \varphi$.

For an MSO sentence φ, consider the following problem $\textsc{Model-Check}_{\varphi,k}$.

Input: A graph G of treewidth $\leqslant k$.

Task: Determine whether $G \models \varphi$.

Theorem 7.1 [1] *For every integer $k \geqslant 1$, for every MSO sentence φ,* $\textsc{Model-Check}_{\varphi,k}$ *can be solved in linear time in the size of the input graph.*

Actually Theorem 7.1 holds even if we allow quantification over sets of edges. For more details, see, for example, [1, 2].

References

[1] B. Courcelle. Graph rewriting: An algebraic and logic approach. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 193–242. 1990.

[2] J. Flum and M. Grohe. *Parameterized Complexity Theory.* Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006.

An algorithm for Vertex-Cover$_k$

Let $G = (V, E)$ be a graph with tree decomposition T of width k. We assume that T is rooted, and that T is binary. We need the following notations.

- For a node X in T, let T_X be the sub-tree of T rooted in X.
- We will write $\bigcup_{Y \in T_X} Y$ to denote the set of *vertices in* G that appears in the tree T_X.
- $G[T_X]$ denotes the subgraph of G restricted to the vertices in $\bigcup_{Y \in T_X} Y$.
- $G[X]$ denotes the subgraph of G restricted to the vertices in X.

Note the difference between $G[X]$ and $G[T_X]$.

For a node X, we will construct $(R_1, N_1), \ldots, (R_m, N_m)$ such that for each (R_i, N_i),

- $R_i \subseteq X$,
- there is a vertex cover $\mathcal{C}$ of $G[T_X]$ such that $\mathcal{C} \cap X = R_i$ and $|\mathcal{C} - R_i| = N_i$. (If there is no such vertex cover $\mathcal{C}$, we let N_i to be ∞.)

Note that since $R_i \subseteq X$, we can fix m to be 2^{k+1}.

The construction is done bottom-up from the leaf nodes of T.

- For a leaf node X, the construction $\ell ab(X)$ is by checking all possible subsets of X, and all $N_i = 0$.
- For an interior node X, we construct the desired $\ell ab(X) = (W_1, K_1), \ldots, (W_m, K_m)$ as follows.

  let Y_1 and Y_2 be the two children of X in T, and let $\ell ab(Y_1) = (R_1, N_1), \ldots, (R_m, N_m)$ and $\ell ab(Y_2) = (S_1, M_1), \ldots, (S_m, M_m)$.

  For each (W_i, K_i), do the following.

  – Determine whether W_i is a vertex cover in $G[X]$. If it is not, we set K_i to be ∞.
  – If W_i is a vertex cover in $G[X]$, we set K_i as follows.

$$K_i \quad := \quad \min \left\{ |(S_j \cup R_l) - W_i| + N_j + M_l \;\middle|\; (S_j \cup R_l) \cap X \subseteq W_i \right\}$$

Since the treewidth is bounded, determining K_i takes constant time. Thus, the overall algorithm takes only linear time in the size of the tree decomposition T, and hence, in the size of G as well.

Now, to solve Vertex-Cover$_k$, we consider the sequence $(R_1, N_1), \ldots, (R_m, N_m)$ in the root node, and check whether there is (R_i, N_i) such that $N_i + |R_i| \leqslant M$.