# Properties of $\mathcal{H}$-clique-width

**Adam Straka**

(joint work with Jakub Balabán and Tomáš Novotný)

Given a class of graphs $\mathcal{H}$, $\mathcal{H}$-clique-width is a generalization of the clique-width. It has a similar definition, but unlike in clique-width expressions, newly created vertices are also assigned a node of a graph $H \in \mathcal{H}$. When creating edges between two colours, we create an edge only if the assigned nodes of its endpoints are an edge in $H$.

The parameter was defined in [1] by Hliněný and Jedelský who showed that a graph $G$ having $\mathcal{H}$-clique-width $k$ is equivalent to $G$ being an induced subgraph of a strong product of some graph from $\mathcal{H}$ and a graph of clique width $k$. In [2] the authors further study the parameter with respect to logic on graphs. They show that the above mentioned product structure is preserved by first-order transductions.

However, the parameter allows for arbitrary choices of $\mathcal{H}$, and there are many open combinatorial questions, when we consider this general setting. What is the complexity of determining the $\mathcal{H}$-clique-width of a graph, can we compute an approximation in polynomial time? Is there a graph of $\mathcal{H}$-cw exactly $k$? For what graph classes $\mathcal{H}, \mathcal{H}'$ is $\mathcal{H}$-clique-width functionally bounded by $\mathcal{H}'$-clique-width?

We show some results regarding the hardness of determining the $\mathcal{H}$-clique-width of a graph. We also study its behavior with respect to taking the complement of a graph. Unlike the clique-width, we show that the $\mathcal{H}$-clique-width does not necessarily behave nicely when taking the complement.

## References

[1] P. Hliněný, J. Jedelský, $\mathcal{H}$-clique-width and a hereditary analogue of product structure, 49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024), LIPIcs 306 (2024) 61:1–61:16.

[2] P. Hliněný, J. Jedelský, Transductions of graph classes admitting product structure, 40th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS) (2025) 843–855.