The graph homomorphism problem in structurally restricted classes

Marta Piecyk

One of the most popular problems in graph theory is k-COLORING. In this problem we are given a graph G and we want to color its vertices with at most k colors so that adjacent vertices get distinct colors (see Figure 1).

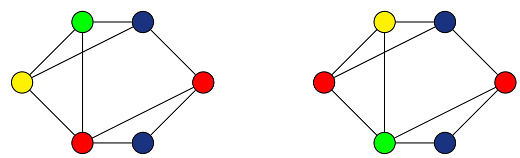

Figure 1: Improperly colored graph (left) and properly colored one (right).

One can try to solve the problem by checking all possible colorings. However, there are exponetially many of them, so such an algorithm is not effective and for large input graph, even modern computers would compute it for years. Therefore, we are interested in algorithms that solve our problems fast, e.g., in polynomial time. In case of k-COLORING it is known that such algorithms exist for $k = 1$ and $k = 2$. This is not the case for $k \geqslant 3$ – we do not know if the problem can be solved in polynomial time. Moreover, it is very likely that such an algorithm does not exist (by famous P$\neq$NP conjecture). However, if we add some restrictions on the input graph G, then the problem can be solved efficiently.

A generalization of k-COLORING is the graph homomorphism problem. In k-COLORING for every pair of ajdacent vertices, allowed pairs of colors are exactly those that are distinct, and in the homomorphism problem the graph H defines which pairs of colors are allowed. More precisely, for a fixed graph H (which we usually treat as a small one) and for a given graph G (that can potentially be large), we want to color the vertices of G with the vertices of H so that the adjacent vertices of G get adjacent vertices of H (see Figure 2).

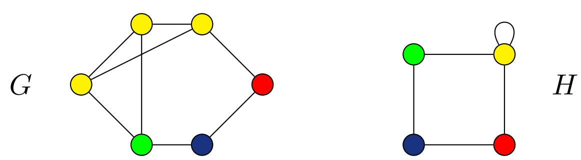

Figure 2: Graph G (left) that is properly colored with the vertices of the graph H (right).

In this project we are interested in the computational complexity of the graph homomorphism problem. We want to investigate how the complexity of the problem depends on the restrictions that we put on the input graph.