Graphs are abstract structures whose role is to model interactions between pairs of objects. Such objects are represented by the vertices of the graph, while the interactions are represented by the *edges* of the graph each connecting some two vertices. In general, the vertices of a graph can represent an arbitrary collection of objects, while the edges can represent an arbitrary relation that can be unambiguously said to be satisfied or not for any pair of objects from the collection. This generality, simplicity of the structures considered, and multitude of applications have made graph theory one of the most important and flourishing branches of finite mathematics.

Geometric intersection graphs constitute a special class of graphs used to model geometric relations of certain types. In such graphs, the vertices represent objects of some geometric space, for example segments on a line (intervals) or in the plane, discs or rectangles in the plane, or balls in the three-dimensional space, while the edges connect pairs of vertices that intersect, that is, have some part in common. Graphs of this kind find practical applications to resource allocation, job scheduling, DNA sequencing, and VLSI design.

In theoretical computer science, computational problems, that is, problems that consist in designing an algorithm to perform some operations on arbitrarily chosen input data, are usually classified as computationally *easy* or *hard*. Easy problems are those that can be solved by a *polynomial-time algorithm*, that is, an algorithm performing at most n^c operations, where n is the size of the input data and c is a constant. Hard problems are those that are expected to have no such algorithm based on widely accepted hypotheses of complexity theory.

The fundamental computational problem for every class of graphs is the *recognition problem*. For graphs with geometric representations, it asks whether an abstract graph given in the input by lists of vertices and edges has a representation appropriate for the class considered. Recognition problems have been considered since the 1970s and for many classes of graphs, they have seen satisfactory solutions—polynomial-time algorithms or computational hardness proofs. All classes of graphs considered in the present project have polynomial-time recognition algorithms. There are also classes of geometric intersection graphs that are hard to recognize, for example, the classes of disc intersection graphs and rectangle intersection graphs.

The present project's research problems are concerned with two generalizations of recognition problems: partial representation extension problems and modification problems.

The partial representation extension problem asks, given a graph and a representation of some part of it, whether that partial representation can be extended to a representation of the whole graph. Particularly interesting is the question for which classes of graphs constructing a representation based on some partial solution is computationally harder that building a representation from scratch. It is worth noting that while it is enough for recognition to find only one representation witnessing that the graph belongs to the class, partial representation extension often requires understanding the structure of all possible representations.

To illustrate the difference between recognition and partial representation extension, we will make use of *planar graphs*, that is, graphs that admit a drawing in the plane with no edge crossings. For the recognition problem, it is irrelevant whether the edges are drawn as straight-line segments or arbitrary curves. Every graph drawn using the latter can also be drawn using the former, and moreover such representations can be constructed in polynomial time. In the partial representation extension problem, only some vertices and edges are drawn in the plane, and the task is to draw the remaining ones creating no crossings. Such a task in a simplified form is often performed by children who attempt to connect two points of a drawing (e.g. a mouse and a piece of cheese) avoiding unnecessary crossings with the lines predrawn. This problem can be solved by a polynomial-time algorithm if the edges can be drawn as arbitrary curves. However, if they are required to be drawn as straight-line segments, the problem becomes computationally hard.

One of the goals of the present project is to find polynomial-time algorithms for or to prove computational hardness of the partial representation extension problem in classes of graphs for which it remains open. The best known among them are the classes of intersection graphs of circular arcs and of trapezoids spanned between two parallel lines. The partial representation extension problem in these classes of graphs is particularly interesting because no description of the structure of all possible representations is known.

The second group of problems considered in this project are *graph modification problems*. They have been considered since the 1970s and still belong to the mainstream of algorithmic graph theory. The modification problem to a class of graphs C asks whether the input graph can be transformed into a graph of the class C by performing at most k operations of some particular kind. Main focus in this project goes to *deletion problems*, in which the only allowed modification operation is to delete a vertex together with all incident edges. Practical applications of modification problems include problems of correcting errors in experimental data. Such an approach is used for example in molecular biology for DNA sequencing.

For all "reasonable" classes of graphs, the vertex deletion problem is computationally hard if the value of the parameter k denoting the maximum number of allowed modifications is part of the input data on equal terms with the graph. However, only input data in which the graph can be huge but the parameter k is small are interesting from the practical point of view. This assumption leads to the notion of an *FPT algorithm*, which given a graph with n vertices and a parameter k, perform at most $f(k)n^c$ operations, where f is some function and c is some constant. Therefore, for small values of the parameter k encountered in practice, FPT algorithms behave like polynomial-time algorithms. In case of parameterized problems, easy problems are those that can be solved by an FPT algorithm.

Modification problems to a class of graphs C require an approach different from the one to partial representation extension problems. For constructing FPT algorithms, one usually makes use of a characterization of the class *C* by so-called *forbidden structures*. Such a characterization asserts that a graph belongs to the class *C* if and only if contains no structure from some list of forbidden structures. FPT algorithms for vertex deletion to the class of graphs *C* must identify all forbidden structures in the input graph and examine their interactions in order to point at most *k* vertices whose removal from the graph will destroy all forbidden structures that it contain.

FPT algorithm for the deletion problem are already known for several classes of geometric intersection graphs, for example, for the class of intersection graphs of intervals on a line. Our primary goal is to find FPT algorithms for or prove computational hardness of vertex deletion to the class of comparability graphs (that is, graphs whose edges represent the relation of two vertices being one less/greater than the other in some order) and the related class of permutation graphs. In the long run, we also plan to investigate the vertex deletion problem to the above-mentioned class of circular arc intersection graphs, for which only very recently the first characterization by forbidden structures has been discovered. We hope that our research on vertex deletion problems will improve our understanding of structural properties of the intersection graphs considered or maybe even give rise to new general techniques of FPT algorithm design.