Computer systems are prevalent in the contemporary society. It is thus of utmost importance that the reliable software can be created and maintained with as low resources cost as possible. One of the possible automations of software development, which could make the process of developing software easier, is automatic synthesis of programs (program synthesis). It is a computational problem in which the aim is finding the program consistent with the user's intent. This intent may take various forms, most common of which are input-output examples and formal specifications.

A search algorithm solving the synthesis problem to be effective must use the knowledge about the problem's properties. The well known No Free Lunch Theorem states that when all problems are considered, then all general purpose search algorithms are equal in terms of efficiency. While there is no globally better general-purpose search algorithm, it may exist for a certain limited class of problems. Using the knowledge about the characteristics of a certain class of problems is crucial for efficient algorithms for it. However, not always such knowledge is known from the start. The goal of this project is to develop methods of automatic acquisition of knowledge of problem's characteristics, and then to design methods to use this knowledge to make the algorithms more effective. This project focuses on the program synthesis algorithms, but the results will most likely go beyond this domain and be widely applicable.

One of the ways to obtain some kind of knowledge about the problem is to use machine learning. For example, an artificial neural network can be used to learn to estimate the features of the expected solution given the user's intent. In the preliminary work, we used such a network to learn the estimated probability of particular instructions given input-output examples. We plan to extend this approach to learning probabilities from formal specifications. Another way of collecting knowledge analyzed in this project is during the solving of a particular synthesis problem, as exemplified for example by collecting counterexamples showing faults of currently generated programs in our other preliminary work.

In summary, the project is designed to increase the knowledge of the ways in which domain knowledge can be acquired and used to facilitate search in the task of program synthesis. The obtained results will most likely generalize beyond those problems, since some of the approaches to be developed in this project will largely abstract from the specifics of program synthesis.

In addition to the impact for fundamental research, this project may have a number of practical implications. Effective program synthesis is a promising avenue for development of more powerful machine learning algorithms, AI systems, and everyday applications.