Computer science plays a fundamental role in the functioning of societies today in virtually all aspects. An important part of this role is the efficiency of software development, which should be as robust as possible and meet the necessary requirements. The practice of software development shows that it is a complex and demanding process, and hence the vision of automating it, even partially, appeared early in the history of computer science developments.

Program synthesis involves generating computer code automatically or semi-automatically based on given specifications or descriptions. This is a very computationally complex problem, and current methods do not handle it sufficiently well. This is because, among other reasons, they construct programs in a way that is significantly different from that of humans.

The standard way for a human to develop software is to use previously acquired knowledge of the programming language. For example, when a programmer is required to write a program that sorts a list of numbers from smallest to largest, he will start by writing a program that implements a more basic function (subroutine), such as finding the smallest number in the list. He will then use this subroutine to write a final program that takes the smallest number in a list of yet-to-be-selected items one by one and adds it to the resulting list. In this way, humans take advantage of the compositional nature of program representations and the ability to create and test subroutines, thus solving smaller problems to arrive at a solution to a larger problem.

Such a way of producing software seems natural to us, whereas program synthesis algorithms are far from operating in this way. In this project, we will analyze deep learning algorithms, which is the most rapidly growing branch of artificial intelligence and machine learning today, for program synthesis that operate in such an iterative manner. We will explore what ways of representing programs for such algorithms are most useful. In addition, we will verify which deep learning models and learning algorithms are best suited to the task of program synthesis.

Research conducted so far in our team indicates that this iterative approach to the synthesis process, formulated as an intertwined formulation of hypotheses and their validation, allows the learning system to better understand the problem and uncover relationships between the various elements of the programs.

This project aims to provide a better understanding of the representations, structures, and algorithms involved in applying deep learning to the problem of program synthesis. The results of the project have the potential to significantly impact the field of program synthesis and extend to other disciplines facing similar challenges, and potentially impact the practice of machine learning, i.e., the use of program synthesis as a form of machine learning.