Reg. No: 2024/53/B/ST6/01620; Principal Investigator: dr hab. Jan Otop

Synthesis via active learning: theory and applications
Abstract for the general public

Computer code is produced in vast amounts every year, and hence making programming easier and more reliable is intensively researched. Recently, there have been developed numerous tools such as ChatGPT, Copilot, Claude and others, which are based on Large Language Models (LLM). These tools are in particular capable of generating computer code from text prompts in a natural language, which makes programming easier. While the results produced by these tools are impressive, LLMs have been criticized for frequent inaccuracies in the generated code. However, the concept of automatically generating computer code is not new.

In 1957 Alonzo Church proposed a process called *synthesis*, in which electronic circuits are automatically generated from a specification given via a logical formula. Though it may seem superficially similar to code generation by LLMs, the underlying idea is very different as the main goal of synthesis is reliability. LLMs generate code from prompts in a natural language, which is imprecise, but easy to use. In contrast, systems resulting synthesis are accurate up to the point that they can be even mathematically proven to be correct. However, this comes at a high computational cost and requires significantly more effort to specify a system unambiguously. Despite these drawbacks, synthesis has been intensively researched for over 60 years because in some areas such as concurrent programs or hardware design, attaining correctness is crucial and difficult.

Approaches to synthesis vary based on the refinement level of a specification. They range from generating complete programs based on an detailed specification, through completing partially written programs to satisfy constraints (a simple specification), to generating programs based on examples with no specification at all, called *programming by example* (PBE). While the synthesis-from-a-specification approach has inspired multiple theoretical research, the PBE approach has been successful in practice. For instance, Microsoft Excel implements FlashFill feature that automatically generates macros from data (just highlight the table with blank cells and press F5).

Specification via examples is an instance of *inductive synthesis*, which is based on *active automata learning*. Active automata learning is a technique of generating *finite-state automata*, which are an abstract model of electronic circuits, by an algorithm that *actively* asks queries and generates the automaton from the answers. This technique, however, has been employed only in applications, in which queries were answered automatically, not manually by humans. The algorithms ask too many queries to answer them manually.

In this research project, we plan to advance theory of synthesis with a particular focus on active automata learning and apply theoretical results to assisted circuit synthesis. We plan to fulfill the following tasks:

Task 1: Complexity of sythesis. We plan to research theoretical questions motivated by synthesis. The obtained results will guide our research in the following tasks by exhibiting promising directions and those which are unlikely to yield success.

Task 2: Active learning under constraints. There are two approaches to synthesis: synthesis from a specification and synthesis from examples. We plan to combine these two approaches and develop a synthesis framework, in which a synthesis algorithm takes a partial specification (constraints) and performs active automata learning under given constraints. We expect that the combination would greatly reduce the number of queries asked by learning algorithms, which would make active automata learning viable for interactive tools.

Task 3: Explainability of synthesis. While the process of synthesis is reliable, there may be errors in the specification or in the answers to the queries. In order to detect problems with the specification, we plan to develop *explainable* synthesis algorithms, for synthesis based on active learning, which produce systems together with information about key decisions made in the synthesis process.

Task 4: Assisted specification of digital systems. Electronic systems are specified using hardware description languages (HDLs) such as Verilog or VHDL. We plan to develop a tool to ease writing HDL code by synthesizing parts of the code using active automata learning approach. In contrast to other approaches, we aim at generating only selected parts in HDLs programs rather than complete systems.