Reg. No: 2017/27/B/ST6/00299; Principal Investigator: dr Jan Otop

# Quantitative specifications: learning, algorithms and applications

Computer systems are ubiquitous nowadays. Despite advances in software engineering and testing, software and hardware still contain bugs. To eliminate all bugs, there was proposed formal verification, in which correctness of software or hardware is proved mathematically. Formal verification is intensively theoretically studied and developed in the industry: Intel uses formal verification in development of new processors; Microsoft and Facebook use formal verification to verify software; Bosch uses formal verification in development of cruise control systems. Among formal verification techniques, the most prominent one is model checking, in which software or hardware are abstracted to simplified structures, called *models*, on which the correctness is checked.

Model checking returns qualitative answers: "yes" or "no". It allows to state questions like "does the server grants every request?" However, a qualitative answer is often insufficient as it does not explain, in our example, how quickly does the server answer. For that reason, we are interested in quantitative aspects, for instance "what is the average response time of the server?" To verify quantitative aspects, there has been recently proposed quantitative model checking, in which one can specify quantitative properties and obtain quantitative answers. Such answers are more informative and allow for better evaluation of the system. Despite numerous advantages, quantitative model checking is not used in practice because of how complicated it is. Specifying quantitative properties is difficult, verification of quantitative properties is computationally difficult and interpretation of quantitative answers is difficult. The goal of this project is to alleviate these difficulties.

The goal of this project is to facilitate quantitative model checking. We focus on the consecutive verification stages: *specification* of quantitative properties, *verification* of quantitative properties, and *interpretation* of answers returned by quantitative model checking.

To facilitate specification of quantitative properties, we will study computational learning of specifications and quantitative specification languages. We will study learning in two settings: *offline*, where the algorithm gets a set of examples and returns a specification consistent with the examples, and *online*, where the algorithm queries the user about values of selected (by the algorithm) words as well as whether the constructed specification satisfies the requirements. Similar learning techniques are successfully applied to qualitative specifications.

Having a quantitative specification, the next step in the verification process is computing the value of this specification on a given model. Models can be very large and hence it is important to develop efficient verification algorithms. In particular, algorithms that operate on simplified models that can fit into computers memory. In this project, we plan to develop efficient algorithms for quantitative model checking.

Similarly to software, specifications can be buggy as well, which may mislead the user about correctness of the system. We plan to develop techniques assisting at validation of the modeling and the specification steps. Such techniques would show the user that theirs specification is, for instance, suspiciously easily satisfied or the answer returned in the verification process does not depend on a large part of the model.