Abstract for the general public
Robust Formalisms for Quantitative Reasoning

IT systems penetrate our lives in various ways and areas. The times of the pandemic have shown that almost every aspect of life, from shopping to medical appointments, can be realized with the use of modern technologies. Despite the continuous development of software engineering, there are still various types of bugs, both in hardware and in software. The way to eliminate them is formal verification, the purpose of which is to create a formal proof of the correctness of the equipment or program. These techniques are used by many commercial companies such as Bosch, Intel, ARM, Microsoft and Facebook.

The verification process can be automated; *automatic verification* is an active field where the properties verified are either qualitative (correct, no deadlock, etc.) or quantitative (performance, resource consumption, etc.). While automated verification covers a wide range of techniques, model verification has proven to be the most practical and versatile technique.

Checking the model consists in building a model of the verified system (*modeling phase*), system property specification (*specification phase*) and verifying that the model complies with the specification (*validation phase*). The response returned in the validation phase can be a logical value for a qualitative property (e.g. the model meets the specification) or a number for a quantitative property (the average number of steps the server must take to respond to a client request). However, it should be remembered that the answer concerns the model, not the modeled system. Its use in the system and its properties depends on the accuracy of the modeling and specification phases.

It is unrealistic to assume ideal modeling and specification accuracy. We need to account for errors in these phases that may result in a model or specification other than intended. However, this is remediable if we require the modeling and specification techniques to be perturbation-proof, ie that the results of the checking phase change in a controlled manner in the event of minor disturbances. Then, even if the model does not exactly match the system, it approximates the quantitative properties of the system. Similarly, perturbation resistant specifications give approximate answers to queries about quantitative properties.

The aim of this project is to develop perturbation-resistant formalisms for the modeling and specification phases, with an emphasis on quantitative specifications. The project is divided into four tasks: the first two are modeling, the third is the specification phase, and the fourth one is based on the database approach. The result of the project should be both the theoretical foundations and tools for reasoning resistant to perturbation.