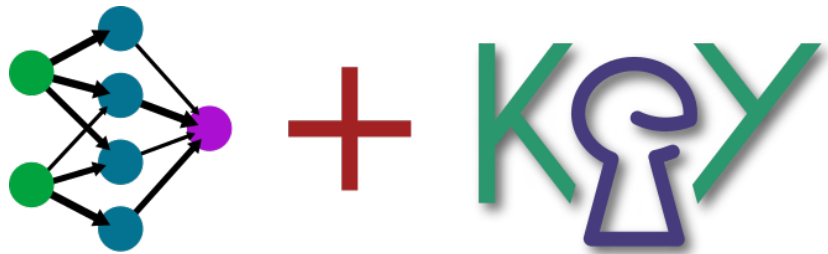## Praxis der Forschung

Using Machine Learning to Improve Strategy Selection in KeY

Deductive program verification is the task of formally proving that a piece of software satisfies its formally specified requirements. This task is undecidable in theory and quite difficult to make work well automatically in practice. The KeY solver is a tool developed in the research group with which Java programs can be verified against a formal specification in a specification language called JML. In the course of the verification process, a variety of intermediate formulas are produced which must be proved. They incorporate different theories (arithmetics, heaps, sequences, sets, user-defined data types, . . .). Currently the automatic prover does not consider the proof situation, but applies the same strategy to all open proof goals. In many cases, that entails a choice that is not optimal; selecting a strategy for every goal tailored to one strategies could have led to shorter or more closed proofs.

Potential. We assume that there is a high potential for machine learning techniques that analyse proof situations and classify them into different proof classes (one might, e.g., be "mainly arithmetic with only few heap assignments").

Your Task. In this thesis, you will ultimately extend the KeY solver with classification step that chooses the right strategy settings for a given proof situation. This includes definition of a suitable *feature set* that discriminiates different proof situations well; and the implementation of the feature extraction in KeY.
You will collect data by running KeY on the regression test set and use different standard ML techniques to construct a classifier. Possible techniques include *decision trees*, *support vector machines* or *neural networks*.

Your profile. Ideally, you have a basic background in *machine learning* and *formal systems* (e.g., from respective lectures of the KIT curricula). You are interested in applying existing ML frameworks (e.g. Scikit Learn) to a new field, and have experience in Java programming to integrate feature extraction into KeY.

Kontakt

Mattias Ulbrich   ulbrich@kit.edu   Office: 50.34, R229