MAX PLANCK INSTITUTE
**FOR SOFTWARE SYSTEMS**

**Debasmita Lohar**

*PhD Student*
*MPI for Software Systems*
+49 (0)681 9303 8112
dlohar@mpi-sws.org
https://people.mpi-sws.org/~dlohar

11 October 2019

## Automatic Verification of Floating-point Rust programs

Numerical computations are widely used in various domains including scientific computing, embedded systems and machine learning. However, as numerical computations are implemented in finite precision (floating-point or fixed-point), the computations may generate exceptions in the code. For example, an operation can produce division-by-zero, square root of negative numbers, floating-point overflow, or underflow. Automatically verifying that these exceptions do not occur is an absolute necessity specifically for safety-critical software systems.

In this project we want to develop a verification technique to guarantee the absence of floating-point exceptions in Rust programs. Rust is a modern systems-level programming language syntactically similar to C and C++ which was designed with safety in mind. Rust was originally developed at Mozilla Research. Currently many companies are using Rust in production all over the world, including Mozilla, Dropbox, Yelp, Braintree and others. The work in this project will get the student acquainted with Rust as well as the basics of static and dynamic analyses of programs which we are developing in our group.

The wider context of the project is to deal with numerical exceptions introduced by approximating real arithmetic with finite-precision arithmetic. There are several static analysis tools that are specifically designed and implemented for detecting floating-point exceptions. However, they do not scale for real-world programs and do not handle all programming language features such as data structures and function calls. This project involves many theoretical, as well as implementation related challenges, ranging from numerical error estimation, to static, and dynamic analyses of programs. In our group, we are working on the theoretical aspects of the problem and are developing a framework that can scale for real-world C programs. Our next goal is to extend it to Rust programs.

The prerequisite of the project is programming experience in C, C++, Java or Python. Experience in Rust is a plus but this is not mandatory. The student will work in a group of international students with computer science background. The group is headed by Dr. Eva Darulova and located in the Max Planck Institute for Software Systems, Saarbrücken, the capital and largest city of the state of Saarland, Germany. Saarbrücken offers a truly pleasant and beautiful locality with France being just around the corner.

The start date of the internship is flexible; it can start between May 15 and July 9, 2020 for a duration of three months (minimum stay ten weeks). In addition to the scholarship and health insurance provided by DAAD, the institute will provide housing free of charge.