**Supervisor:** Viorel Preoteasa and Stavros Tripakis

**Contact information:** viorel.preoteasa@aalto.fi and stavros.tripakis@aalto.fi

**Job description:** Invariant based programming is a technique for constructing correct programs. In invariant based programs the invariants (properties that the program should satisfy) are introduced first, and then the program transitions are introduced such that they preserve the invariants. For example a program that searches for an element $x$ in some array $a[0]...a[n-1]$ starts from a situation where some invariant (the precondition) is true, and if the element $x$ is found then it ends in a situation (a post-condition) where $i$ is the index in the array such that $a(i) = x$. If the element is not found in the array, then the program ends in a situation in which for all indexes $i$ between 0 and $n-1$, $x$ is not equal to $a(i)$. To be able to use invariant based programming in practice we need tool support. The purpose of this project is to build support for invariant based programs in Isabelle/HOL theorem prover. With this tool it will be possible to construct programs, verify (using Isabelle) that the transitions preserve the invariants, and generate executable code (SML, Scala, ...). The work on this project would involve SML (functional) programming and small theory development, as well as becoming familiar with the Isabelle theorem prover. Experience in programming is required for this project.

**Additional information:** [Verification and code generation for invariant diagrams in Isabelle](https://link-to-journal). Journal of Logical and Algebraic Methods in Programming