# Static Type Checking for Jolie Programming Language

## 1. Goals of the Project

The goal of the project is to develop a SMT system for static type checking for Jolie programming language. The resulting system should be able to automatically verify correctness of a program source code in terms of type consistency.

## 2. Overview of the System Specification

The main functionality of program verifying is carried out by Z3. Z3 is a tool from Microsoft Research for checking satisfiability of logical formulas. Given a program written in Jolie, the system will transform it into a set of logical formulas according to its structure, performed operations and declarations. The resulting set of formulas will then be processed by Z3, returning whether it is satisfiable or not.

The system is an extension of Jolie. The type checker is written in Java and uses the same parser of Jolie source code as Jolie interpreter. Therefore, all the work is performed over AST constructed from a source code, and its nodes.

## 3. Background Theory

The essential publication behind the type checker is “A Type System for the Jolie Language” written by Julie Meinicke Nielsen, Technical University of Denmark. Also, the common theories apply: predicate logic, graph theory (abstract syntax tree), satisfiability modulo theories, static type checking.

## 4. Overview of Task Specification and Project Schedule

In general, the list of tasks required for the project consist of tasks for creating handlers for each type of node of Jolie AST. Full list of node types can be found here: <https://github.com/innopolis-jolie-smt-typechecker/jolie/tree/master/libjolie/src/jolie/lang/parse/ast>.