Search for Algorithmic Errors in the Program Code Using Machine Learning Methods

Vasilchenko, Dmitrii (School: Lyceum 1533 of Information Technologies)

Existing static code analyzers are aimed at finding errors (syntax and semantic) for which there are explicit formal criteria, for example division by zero. They are capable of detecting only a few simple types of algorithmic errors. At the same time, significant progress has been achieved in the analysis of natural language texts using machine learning methods. It looks promising to apply machine learning for analyzing algorithmic errors in the source code of the program. The primary goal of this project is to create a tool for finding swapped variables in the source code of C# programs. The simplest example would be calling of print(y,x) while in other parts of the program they are used as print(x,y). As a more general task the tool should propose replacement for misused variable in a given slot. This is also known as VarMisuse problem. To accomplish the task we apply machine learning algorithms for the abstract syntax tree (AST) of the source C# code, augmented with data flow edges. The process could be divided in the following stages: 1. Building a graph from the source code 2. Generation of input data for the neural network 3. Running neural network to find inconsistent data flow 4. Mapping output findings to the original source code of the program. It was tested on various opensource projects with randomly injected algorithmic errors. Depending on configurations of neural network it recognized 80-90% cases of swapped variables. The proposed approach looks promising for finding algorithmic errors in a more general form, such as using of one variable instead of another. This is a new uncharted area of code analysis. The authors would like to contribute to this area as it looks helpful and highly demanded.