

# Genetic Improvement using Grammars

Michal Wiglasz

[iwiglasz@fit.vutbr.cz](mailto:iwiglasz@fit.vutbr.cz)

October 15, 2015

Genetic programming (GP) is a machine learning method inspired by Darwinian evolution, which automatically generates whole programs in given programming language. They involve a population of individuals, each representing one candidate program. Each program is given a fitness value, which is higher if the program's behaviour is closer to the desired functionality. The fitness and parent selection functions represent the “survival-of-the-fittest” selection pressure known from the nature.

Grammars are core representation structures in Computer Science. They are widely used to represent restrictions on general domains, limiting the expressions that may be used. They can be used to define the legal expressions of a computer language, to impose type restrictions, or to describe constraints on interactions within systems. As such, they can be naturally used in genetic programming.

Genetic improvement (GI) uses genetic programming to find somehow improved version of existing human-created software. The aim of GI can be for example better performance, lower memory requirements, bug fixing or porting code to different platform. The source code can be used as the template for its own improvement, in which case it is analyzed to create a special “one-sentence” BNF grammar. In this grammar, rules correspond to single lines of source code, conditional parts of `if`, `else` and `while` statements, and the initial, increment and test parts of the `for` statement. The evolution is then constrained by the grammar, which ensures that the code is always syntactically correct and that structures like classes, functions or data structures are retained. GP modifies the grammar by exchanging rules of the same type (for example two `if` conditions), yielding syntactically valid – and hopefully better performing – program.

This work deals with method to construct BNF grammar from existing source code and properties of the resulting grammar. It also describes the genetic programming itself and genetic operators used to modify the grammar during the search for improved version of the analyzed program.