Results of metric based transformations

Roland Király

Eszterhazy Karoly University of Applied Sciences
kiraly.roland@aries.ektf.hu

Abstract

In this paper I am going to present how to create an analyzer that reveals inadequate programming style or overcomplicated erlang [8, 9] program constructs during the whole lifecycle of the code using complexity measures describing the program.

The algorithm [4, 6, 7], which I present here is also based upon the analysis of the semantic graph built from the source code (which, in this case, is naturally based on the measurements of structural complexity), but at this stage we can define default complexity measures, and these defaults are compared to the actual measured values of the code, and so the differences can be indicated.

The impact analysis of the transformations is theoretical but based on conclusions, as these statements are proven by running tests and interpreting the results on the implemented analyzing prototype.

On the other hand I show the algorithm measuring code complexity in Erlang programs, that provides automatic code transformations based on these measures. I created a script language that can calculate the structural complexity of Erlang source codes, and based on the resulting outcome providing the descriptions of transformational steps. With the help of this language we can describe automatic code transformations based on code complexity measurements.

I define the syntax [5] of the language that can describe those series of steps in these automatic code refactoring that are complexity measurement [2, 3] based, and present the principle of operation of the analyzer and runtime providing algorithm. Besides the introduction of the syntax and use cases, I present the results we can achieve using this language.

After defining and implementing the language [12, 13, 14], I create several scripts solving source code transformational problems that are based on complexity measures, and present their run-time results, proving their usability.

The transformation scripts created by me were applied on the source code of large programs, too. In this paper I present the results I measured during

---

*This research was supported by the European Union and the State of Hungary, co-financed by the European Social Fund in the framework of TÁMOP 4.2.4. A/2-11-1-2012-0001 ‘National Excellence Program’.
the run time of transformation scripts to prove the validity of the analysing algorithm and language.

Keywords: software metrics, complexity metrics, source code, semantic graph, refactorerl

References


