scalafmt: automatic, opinionated code-formatting for Scala.

Ólafur Páll Geirsson

School of Computer and Communication Sciences

Semester Project

June 2015

Responsible
Prof. Martin Odersky
EPFL / LAMP

Supervisor
Eugene Burmako
EPFL / LAMP

Abstract

Automatic code formatters bring a lot of benefits to software development. When done right, code formatters relieve the developer's attention from syntactic trivia and enforce a consistent coding style between developers. However, developing a good code formatter is still somewhat of a black art and yet, little research has been made towards the algorithms that power such code formatters.

This thesis addresses the problem of developing a feature-rich code formatter for a custom programming language. Our contributions are twofold. First, we present a language agnostic framework that consists of core data structures, algorithms and tooling that allow rapid development and testing of such a code formatter. Secondly, we use our model in a case study and provide a concrete implementation of a code formatter, `scalafmt`. We show that the framework makes

Contents

1 Introduction

The main contributions presented in this thesis are the following:

- A DSL for declaring feature-rich code formatting rules.
- An case study where the DSL is used to format Scala programs.

Throughout the paper we assume familiarity with the basics of the Scala Programming Language [1].

2 Framework

2.1 Data structures

2.2 Interpreter

2.3 Tooling

3 scalafmt

3.1 Data structures

3.2 Interpreter

3.3 Tooling

4 Conclusion

References

[1] Martin Odersky et al. *The Scala language specification*. 2004. URL: http://www-dev.scala-lang.org/old/sites/default/files/linuxsoft_archives/docu/files/ScalaReference.pdf (visited on 05/31/2015).