

JULIA in a Nutshell

Fausto Spoto

Università di Verona & Julia srl

`fausto.spoto@univr.it`

Abstract

This is a very gentle introduction to the JULIA software for static analysis, meant for programmers and system managers. We describe what it allows one to do and how one can use it inside his organization.

JULIA is a purely Java program for static analysis of other Java or Android programs. Its goal is to find programming bugs, automatically and without any help on the part of the programmer. It uses the most modern techniques for static analysis based on formal methods such as abstract interpretation. As a consequence, it comes with a correctness guarantee: if there is a bug of a class considered by JULIA, then JULIA will find it. Since finding all and only the bugs goes against the basic laws of the computability theory, our tool has to issue spurious warnings, together with warnings that are actual bugs. Special care has been taken to reduce the number of the spurious warnings, without sacrificing the correctness guarantee.

Currently, JULIA looks for the following bugs:

- dead-code** Methods or constructors that are not called: they can be eliminated or are actual bugs;
- incorrect casts** A typical programming error, when the programmer assumes that some data has a different nature than it actually has;
- null-pointer errors** Another typical programming error, when the programmer accesses data that is actually missing;
- non-terminating loops or recursion** They are a waste of computing resources or an actual bug;
- bad style** Inappropriate class or method names; access of static data from a non-static context, *etc.*;
- unused fields** Fields that are only read or only written are useless and can be removed, or can be an actual bug;
- wrong redefinitions** Inappropriate types when a method is redefined, or inappropriate calls to `super()`;

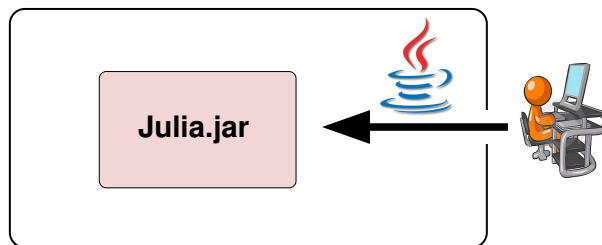
bad equalities Use of `equals()` instead of `==` or vice versa, use of `equals()` on arrays

bad definitions of equals/hashCode Missing `hashCode()` or `equals()`

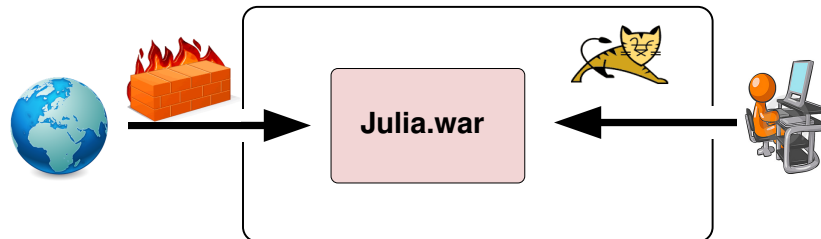
bad comparisons Unsafe, approximated comparisons between float numbers.

More kinds of bugs are being added continuously and we are happy to add new static analyses to check specific features of software, relevant for our client software companies. JULIA sits on a complex and very precise set of supporting analyses for approximating the heap memory. New analyses can be easily built upon them and exploit their full power.

Programmers can query JULIA to find bugs in their software or in third-party software. Our tool only needs the compiled bytecode: it does not make any use of the source code. Programmers can use JULIA as a traditional command-line tool, run by any standard Java Virtual Machine, overnight or before any important release build, as the following picture shows:



A specific feature of JULIA, that makes it completely different from other software analysis tools, is that it can be queried as a web service, run inside a standard Tomcat container, as in the following picture:



In this second scenario, access to JULIA is granted from the local network, which is useful for a software company that provides access to JULIA to all its programmers. But it is also possible to access JULIA from the external world, as a remote Internet web service, as long as the company firewall allows access to our tool. Access to this web service is provided in the SOAP style, perfect for batch processing; but also through a flash web interface. An Eclipse plug-in that connects to the web service is on our to-do stack.