Contribution ID: **22** Type: **Talk (15min + 5min)**

Assisting Data Analysis using Program Slicing with flowR

Wednesday 26 February 2025 12:00 (20 minutes)

Consider you are a reviewer checking the correctness of a research artifact or a data scientist searching for a data cleaning step or visualization to reuse.
Either way you are confronted with hundreds of lines of code, usually involving various datasets and several different plots, making it difficult to understand the code's purpose and the data flow within the program.
Addressing this issue, program slicing reduces scripts to only what is relevant for a plot or data transformation.
Furthermore, it assists authors when writing code, indicating parts which are not relevant for the desired output and helping them to improve the analysis structure.

With our talk, we introduce flowR, a novel program slicer and dataflow analyzer for the R programming language.
Given a variable of interest, like a plot, flowR returns the resulting slice either directly or interactively within an IDE.
We provide a flowR addin for RStudio and a more feature-rich extension for Visual Studio Code which will be the focus of the talk, offering features like multi-cursor support, highlighting, and more.
Additionally, we provide a server session, a read-eval-print loop, and a GitHub Codespace to try flowR without any installation.
flowR is developed as an open-source project (under the GPL-3.0 license) on GitHub and offers a docker image.

We focus on R, because the set of existing tools to support the large and active community is relatively small, without any preexisting program slicer.
Although the RStudio IDE and the R language server, as well as the {lintr} and the {CodeDepends} package perform static analysis on R code, all of these tools
rely on simple heuristics like XPath expressions on the abstract syntax tree (AST), causing their results to be imprecise, limited, and sometimes wrong.

flowR first normalizes the AST, using it as the basis for a stateful fold, incrementally constructing the dataflow information of each subtree. For the analysis, we use a dynamic dispatch on an abstract interpretation of the active R environment to handle the language's dynamic nature. With the dataflow graph, the program slicing reduces to a reachability problem solved by a modified breadth first search. Finally, the slice is either reconstructed as R code or highlighted directly in the input.

Currently, we offer limited support for R's wide variety of side effects (e.g., modifying functions at runtime), defaulting to over-approximation and conservative results.
However, we use automatic input-output validation on a wide set of sources to make sure that the generated slices and the dataflow graph are correct.

Using real-world R code (written by social scientists and package authors) shows that flowR can calculate the dataflow graph (and the respective slice for a given variable reference, including the parsing and normalization) in an average of 200-500ms. Slicing for manually selected points of interest (e.g. plots), we reduce the program on average to just 12.7%[±11%] of its original size.

In our talk, we focus especially on flowR's extensions and how they benefit R programmers in their everyday work.

I want to participate in the youngRSE prize

yes

Primary authors: SIHLER, Florian (Ulm University); Prof. TICHY, Matthias (Ulm University)

Presenter: SIHLER, Florian (Ulm University)

Session Classification: Workflows for data pipelines

Track Classification: Data and Software Management: computational workflows